PRI : 10

A Publication

FACIAL RECOGNITION PROJECT REPORT

Issued by:

W. W. BLEDSOE

PANORAMIC
RESEARCH
INC

3946 Fabian Way, Palo Alto, California — DAvenport 1-6592

FACIAL RECOGNITION PROJECT

REPORT

Submitted To

KING-HURLEY RESEARCH GROUP
422 Washington Building
Washington, D.C.

Submitted By

DR. W. W. BLEDSOE
PANORAMIC RESEARCH, INC.
3946 Fabian Way
Palo Alto, California

6 March 1964

TABLE OF CONTENTS

Facial Recognition Project Report
W. W. Bledsoe

Appendix I - A Report on Picture Scanning and Processing by Machine
Selden L. Stewart III.

Appendix II - Computational Method for Determining Eye Location on a Transduced Picture - Charles K. Williams

Appendix III - Reconstructed Pictures
W. W. Bledsoe

Appendix IV - Rotation of Faces
Helen Chan, Lawrence Cowdery and W. W. Bledsoe

Appendix V - Feature Extraction Method for Recognizing Faces
W. W. Bledsoe

Appendix VI - A Report on Machine Reading of Handwritten and Printed Characters - C. L. Bisson

Appendix VII - A Corridor-Projection Method for Determining Orthogonal Hyperplanes for Pattern Recognition - W. W. Bledsoe

Appendix VIII - PRI Memorandum - Optimization of n-tuple memory matrices using a gradient technique - C. Bisson

Appendix IX - The Generation of Variable Characters
W. W. Bledsoe

"Classification and Generalization Capabilities of Linear Threshold Units", Thomas M. Cover, Stanford Research Institute, November 1963

"Improved Memory Matrices for the n-tuple Pattern Recognition Method," W. W. Bledsoe and Charles L. Bisson, IRE Trans. on Elec. Comp., June 1962, pg. 414-415.

"Pattern Recognition and Reading by Machine," W. W. Bledsoe and I. Browning, 1959 Proceedings of the Eastern Joint Computer Conference.

Proposal for A Study to Determine the Feasibility of a Simplified Face Recognition Machine - Submitted by: W. W. Bledsoe, PRI.

Bibliography

FACIAL RECOGNITION PROJECT

Report

A. AUTHORIZATION

This project was undertaken under contract PR-1 from the King-Hurley Research Group, Washington, D. C., dated 28 February 1963 (effective 1 January 1963) and extended.

B. PURPOSE

The general purpose of this research was to determine the feasibility of recognizing pictures by machine. See "Proposal for a Study to Determine the Feasibility of a Simplified Face Recognition Machine," attached herewith.

C. PROBLEM

Can a machine be designed which will, when given a facial photograph of a person, determine the identity of that person? More specifically, suppose the consideration is restricted to a fixed total group of Q people where Q is large.(For example, Q might be larger than one million.) If the machine is given at least one identified facial photograph of each person in the group during a "learning" stage, can it later identify another photograph of one of the people in the group? Or, more realistically, can it select out of the total group a small subgroup (the answer group) which contains the person who's identity is sought? If the number, A, of people in the answer group is small then the machine operation could be followed by a human search to determine final recognition. It is thus the purpose of the proposed machine to reduce the total group, which is too large for humans to fathom, to an answer group which can be easily handled.

In the earlier stages of research, the machine referred to in the above paragraphs might well be a combination of an optical scanner and a digital computer. In this case the programs for the digital computer which effect the proposed recognition are in reality the "recognition machine."

D. PRESENT STATE OF THE ART

No such machine now exists to satisfactorily solve this problem. Some attempts have been made for very small groups of people and very little variability in poses. In fact, in the research that has been done so far, the machine is usually asked to recognize the exact photographs that it has learned. However, parts of the ultimate problem have been solved.

E. DIFFICULTIES

Why is this problem not the same as the problem of recognizing letters of the alphabet, say in printed or handwritten form? The answer is given in two parts.

1. The number Q of categories, or possible identities, is much larger in the facial case. For example the total group of people considered might be 1,000,000, whereas the total number of letters of the alphabet is only 26, or still less than 100 even when upper and lower case, numerals, and punctuation marks are included.

2. The variability of the image for the same category is usually much greater for faces than for letters because of three dimensional rotations, changes in lighting intensity and direction, changes in expression, physical changes such as growth of hair and ageing, etc.

F. POSSIBLE SOLUTION

It would appear that the solution to the facial recognition problem would lie in one or more of the following:

1. Developing more powerful recognition techniques for identifying the whole face at once.

2. Dividing up the face into a sequence of parts or features, recognizing (or classifying) each of these features, and combining the parts somehow for a recognition of the whole.

3. Partitioning the categories in such a way that at any one time the machine is required to make decisions between only two (or a few) super categories. For example, the machine might first decide whether the person is male or female, then decide whether the person is Oriental or not, etc.

4. Preprocessing of the pictures to eliminate non-essential variability before the recognition process began.

The present study followed all of these lines of attack. Also substudies were carried out on methods and techniques that could be used as parts of an overall system.

The following is an outline of the work performed on this study. Detailed descriptions of this work are given in the several appendixes which follow and which are referenced below.

G. OUTLINE OF STUDIES MADE

1. Photographic Procedures (Appendix I)
    a. For general input
    b. For facial rotation study (Appendix IV)

2. Optical Scanning
    a. Equipment developed and used - what does it do
    b. Data available on magnetic tape for future use
    c. Future possibilities

3. Preprocessing of photographs after they have been scanned.
    a. Reconstruction of photographs (Appendix III)
    b. Scaling of size and normalization of lighting intensity (Appendix I)
    c. Facial rotation (Appendix IV)
    d. Masks (Appendix I)
        (1) Type 1
        (2) Type 2
        (3) Data available on tape
        (4) Other possibilities

4. Recognition and partitioning of categories.
    a. Hyperplane study (Appendix I)
    b. Corridor-projection method (Appendix VII)

5. Feature extraction method of recognition. (Appendix V)
    a. Location of features
        (1) "Find eye" program (Appendix II)
        (2) Other techniques
    b. Recognition of features (Appendix VI)
        (1) Results on handwriting
        (2) Label printing study
    c. Combination of feature classifications.

H. DISCUSSION AND CONCLUSIONS

The picture normalization problem is formitable and perhaps unsolvable at this time. The major difficulty being the many poses that one person can present to a camera. These include variations in: light intensity, light direction (shadowing), distance from camera (definition), rotation of head (2-dimentional), facial expression, hair growth, ageing, background, etc. Unless this tremendous source of variability can be substantially reduced, the facial recognition problem seems beyond the capacities of present day methods and equipment.

Some of these, such as lighting, definition, scaling, and background, present no particular difficulty (See Appendix I), but those like hair growth, expression, and ageing are difficult indeed. This study made a serious attempt to eliminate the rotation-of-head problem with some success (See Appendix IV); however, even here the outcome seems uncertain unless some kind of stereographic process can be utilized.

It is recommended that continued research be conducted on these type problems, at a low level but over many years. It is not felt that a large concerted effort now would eliminate these normalization difficulties in a short period of time.

Of course, a completely new approach may partially remove the requirement for normalization. To some extent this is what is attempted in Appendix V, "Feature Extraction Method for Recognizing Faces". Parts of this approach were thoroughly studied (Appendix VI) but the overall system proposed there was not tried or even formulated in detail. This lack of completion was due partly to the ending of the contract and partly to the inherent difficulty of the problem. Here too, it is recommended that the research proceed at a low rate over an extended period of time.

Appendix VII presents an entirely new approach to the discrimination problem when there are large number of possible categories (for example, a large number of people to be recognized). This method was tried with success on some simple and moderately difficult recognition problems, but has yet to be pitted against the formidable facial recognition problem. Its chances of success can be greatly enhanced if it can be preceded by some normalization as indicated above.

A recent study by Cover and Efran[*] has given evidence which seems to infer that the size of a computer necessary to solve a recognition problem, crudely speaking, is portional to the difficulty of the problem. For example, in the picture recognition problem we might need a computer with several times as many storage elements as there are people to be recognized. In the case of a million people this might amount to billions of storage elements unless some effective normalization techniques can first be applied. It will be some time before the real effect of Cover's important paper can be determined.

It is concluded that picture recognition by machine for a large sample of people is beyond the state of the art of the present pattern recognition and computer technology at this time.

I. RECOMMENDATIONS

It is recommendated that low level long range support be given for research in picture recognition by machine, so that continued efforts can be made to overcome the difficulties inherent in this problem. It is further recommended that limited short range support be given to complete certain studies at PRI which were interrupted before completion and for which data has been collected. These include:

1. Facial rotation.
2. Determining three-dimensional structure of a face.
3. Determining the effectiveness of the Corridor-Projection Method on a small sized facial recognition problem.

PRI has several hundred photographs on film, some of which have been scanned and transduced to magnetic tape, which can be used for this purpose.

W W Bledsoe

W. W. Bledsoe

---

[*] T. M. Cover, "Classification and Generization Capabilities of Linear Threshold Units," Stanford Research Institute, Nov. 1963. (Copy attached to the report.)

APPENDIX I

A REPORT ON PICTURE SCANNING AND PROCESSING BY MACHINE

Selden L. Stewart III

1. Introduction.

The pictures being processed here, facial photograph of humans, were made and developed upon 16 mm movie film (See Figure 1). These films were then processed by an optical scanner and digital computer as described below.

Procedural Outline:

The entire recognition process from photographic transparancy to identification can be outlined in four steps.

(1) Optical scanning of the photographic material.

(2) Preprocessing the results from step 1 so that each picture is in a standard format on high density magnetic tape.

(3) Converting the picture in standard format to a picture vector by anyone of several abstraction programs.

(4) Recognizing the individual from the picture vector.

It is not necessary that the steps be physically separable; although that is the case in the initial implementation. Once the major problems in system design have been solved, the first three steps, if not all four, would probably be consolidated in one piece of hardware. Both the original version of the scanner and the recent improvements will be described. The data used in the experiments thus far have come from the earlier model scanner, but the new model will significantly improve the input data and combine steps 1 and 2 above. Next will be an

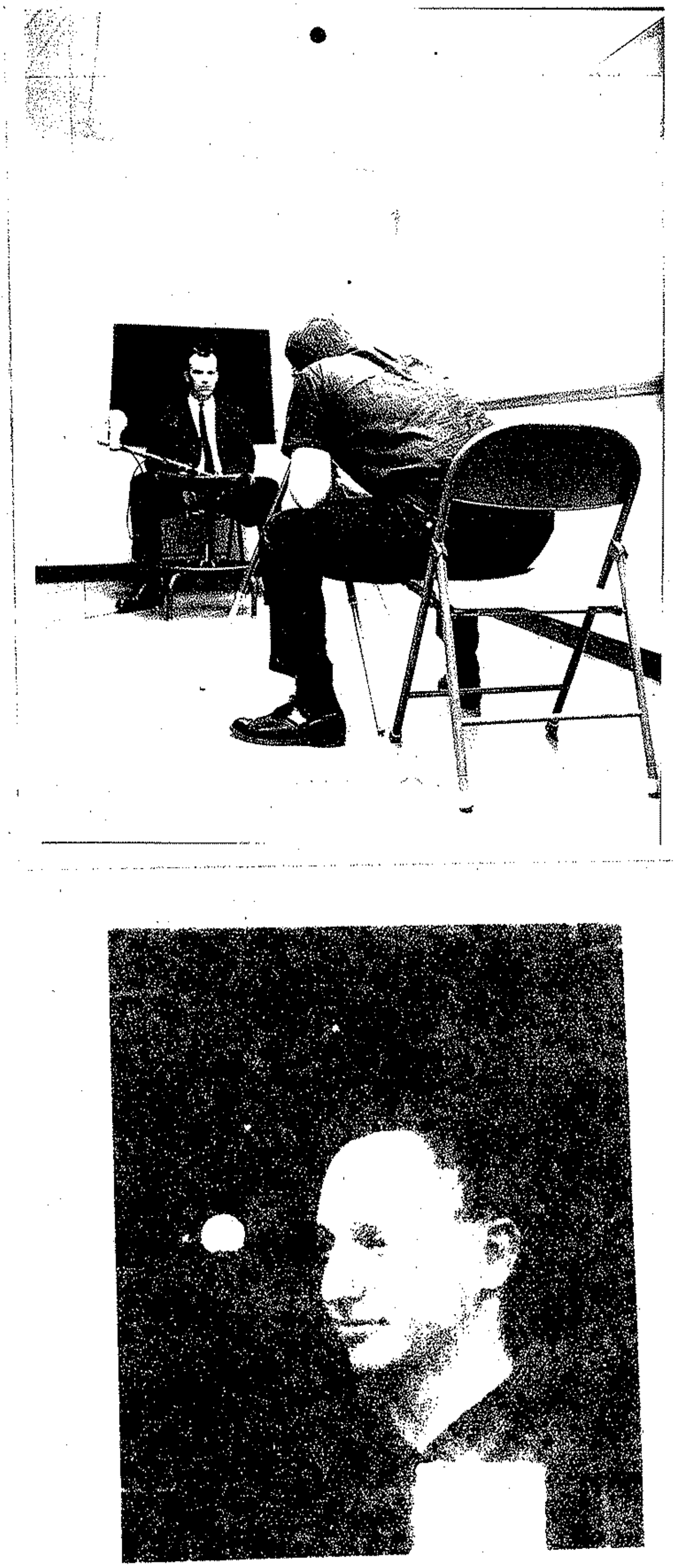


Figure [illegible]

The Photographic Procedure

explanation of the preprocessing step for the original scanner. Then there will be a discussion of the problems of steps 3 and 4. And finally two of the programs used in the last two steps.

2. Optical Scanning Device[1].

The initial stage in any system which uses digital computer techniques is to convert data into a form acceptable to the computer, i. e. digitize the data. In our problem, the original data are frames of 16 mm movie film. There is no unique way of digitizing a picture, but there is a very general solution which lends itself to hardware implementation. We determine the amount of light at each point of a fine, rectangular grid superimposed on the picture. If the amount of light at each point is measured by the voltage level of a photomultiplier output, then we can represent the picture by an array of numbers where each number is proportional to the voltage level at the corresponding point. The result is like an electronic half-toning process.

Data processed to date have been done on a scanner adapted for use with an IBM 729II tape drive. Fig. 2 is a schematic of the scanner. The cathode ray tube (CRT) generates a line of points of light at a rate compatible with the tape input (15,000 points/second). These spots of light are imaged on the film by the lens system. The light which passes through the film is collected by a single lens and focused on the photomultiplier. The voltage level from the photomultiplier is clipped at a preset value. If the voltage is above this value, a "1" bit is transmitted to the tape drive. The line could be scanned any number of times using a different clip level each time. In this scanner the circuitry is set for three clip levels. Therefore each point can be assigned one of four light intensity values: below all clip levels, below two clip levels, below one clip level, or below none. The result is a magnetic tape in the format of Figure 3 (schematicly).

---

[1] The scanning device used here was developed by I. Browning.

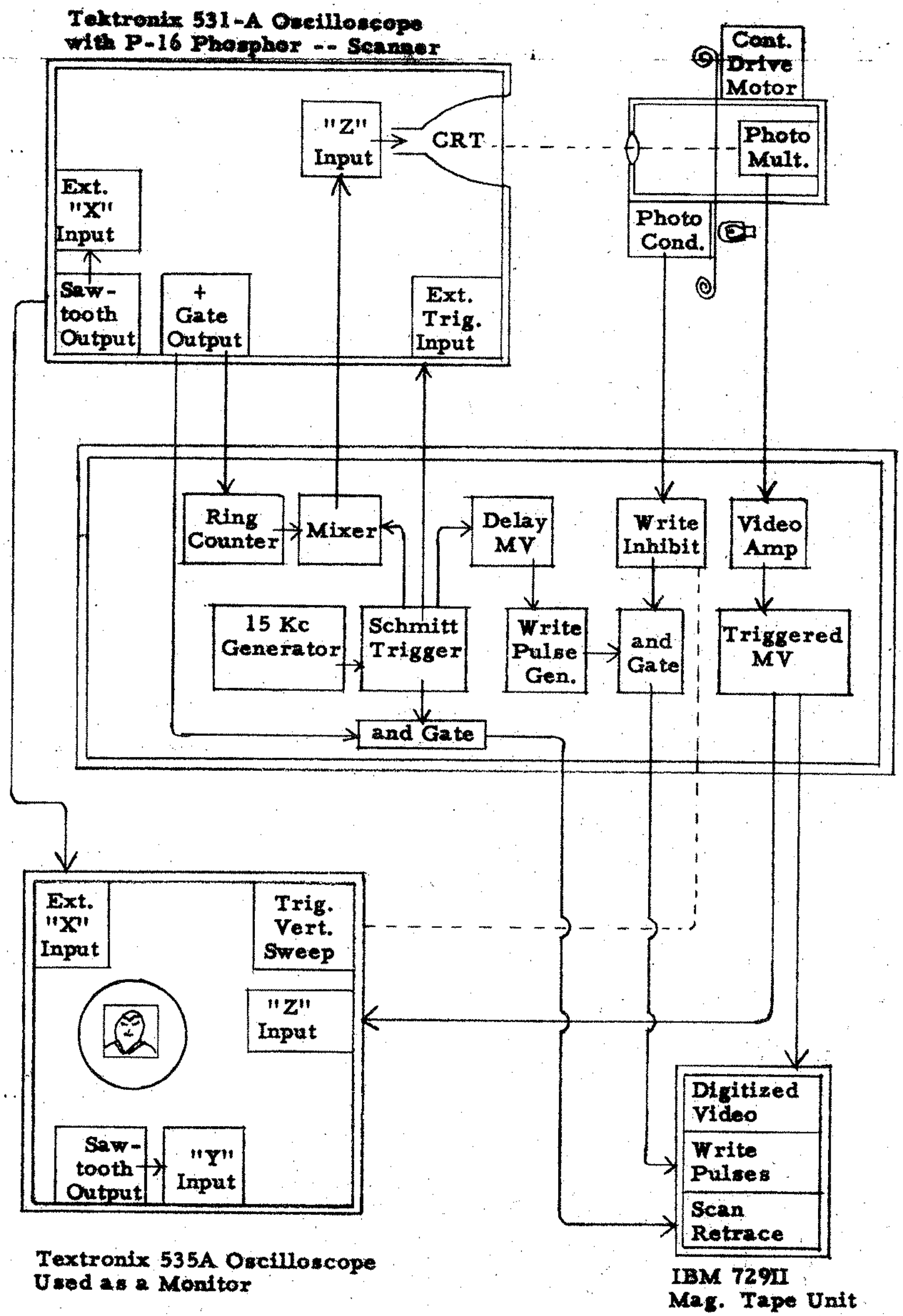


Figure 2

Schematic Drawing of Scanner

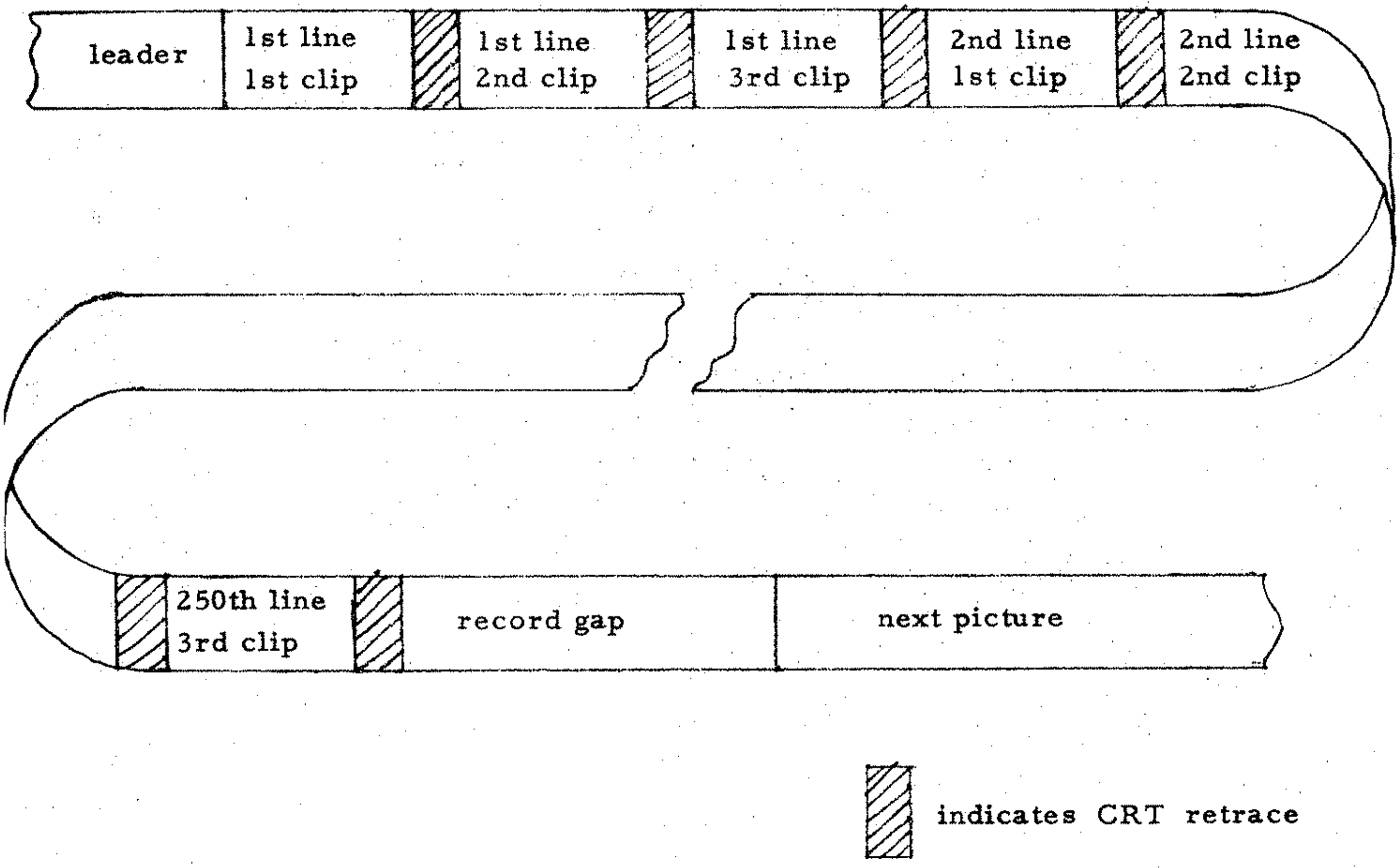


Figure 3

Magnetic Tape Format

3. Preprocessing of Images

A tape with data on it in the form of Figure 3 is used as an input tape for a CDC 1604 program called SUM3 which:

(1) Eliminates the retraces.
(2) Collects the clip levels and assigns from 0 to 3 to represent the light intensity at each point.
(3) Packs all these numbers into one record and writes this record on a high density output tape. (The input tape contains 26 to 28 pictures per 2400' reel. The output tape may store as many as 800 pictures).
(4) As an option it will print the picture on a line printer for checking purposes.

The SUM3 program and the output tape are arranged in such a way that the program may be run again and again with different input tapes to keep adding pictures to the output tape.

We have on hand 120 pictures in the high density form. There are 80 more in the low density form ready to be preprocessed. In addition there are that many more pictures on unscanned film.

4. Current Work on the Optical Scanner.

Recent development of the scanner is directed toward producing a condensed tape more directly. The picture is scanned in much more detail and with a wider range of light intensity values. This new scanner will resolve the picture into 120,000 points with 64 levels of light intensity. The earlier scanner resolved 52, 900 points with 4 levels. The scanner itself is essentially the same in both models. The difference is primarily in the associated circuitry. With both sets of circuitry the earlier model is still available with a minimum amount of conversion.

In Figure 4 we have a schematic of the new scanner system. The part labeled "Scanner" corresponds to the upper third of Figure 2. The output voltage of the scanner is passed through an analogue to digital converter which is directly connected to a CDC 160-A. Both the converter and the scanner are under program control from the computer. In this way the computer, through the program, carries out many of the functions of the circuitry in the middle third of Figure 2. At the same time the program does the preprocessing and writes a high density output tape.

5. Scaling and Normalizing Pictures.

With picture in digitized form, there are several real or potential problems to be faced. (1) Film density is not uniform from one piece of film to another. (2) Position of the face may not be constant from one frame to another. (3) There may be variations in the voltage level of the scanner.

In a sense any scaling or normalization of the picture is going to result in a loss of information. Some of the information in the picture may be irrelavent to the recognition problem; for example, changes in film density are independent of whose picture is taken. Eliminating such information from the picture will generally speed up the recognition process because there is no time wasted in checking these unimportant variables. Also, in the next section we shall see that such information must be thrown away to bring the problem down to manageable size. Therefore it is more reasonable to start by eliminating that which is least useful.

One of the most widely known normalizing transformations is to subtract out the mean and divide by the standard derivation. Suppose the digitized picture is represented by the matrix of light values $\left[x_{ij}\right]$ where x_{ij} is the digitized value in the i^{th} row and j^{th} column. Then

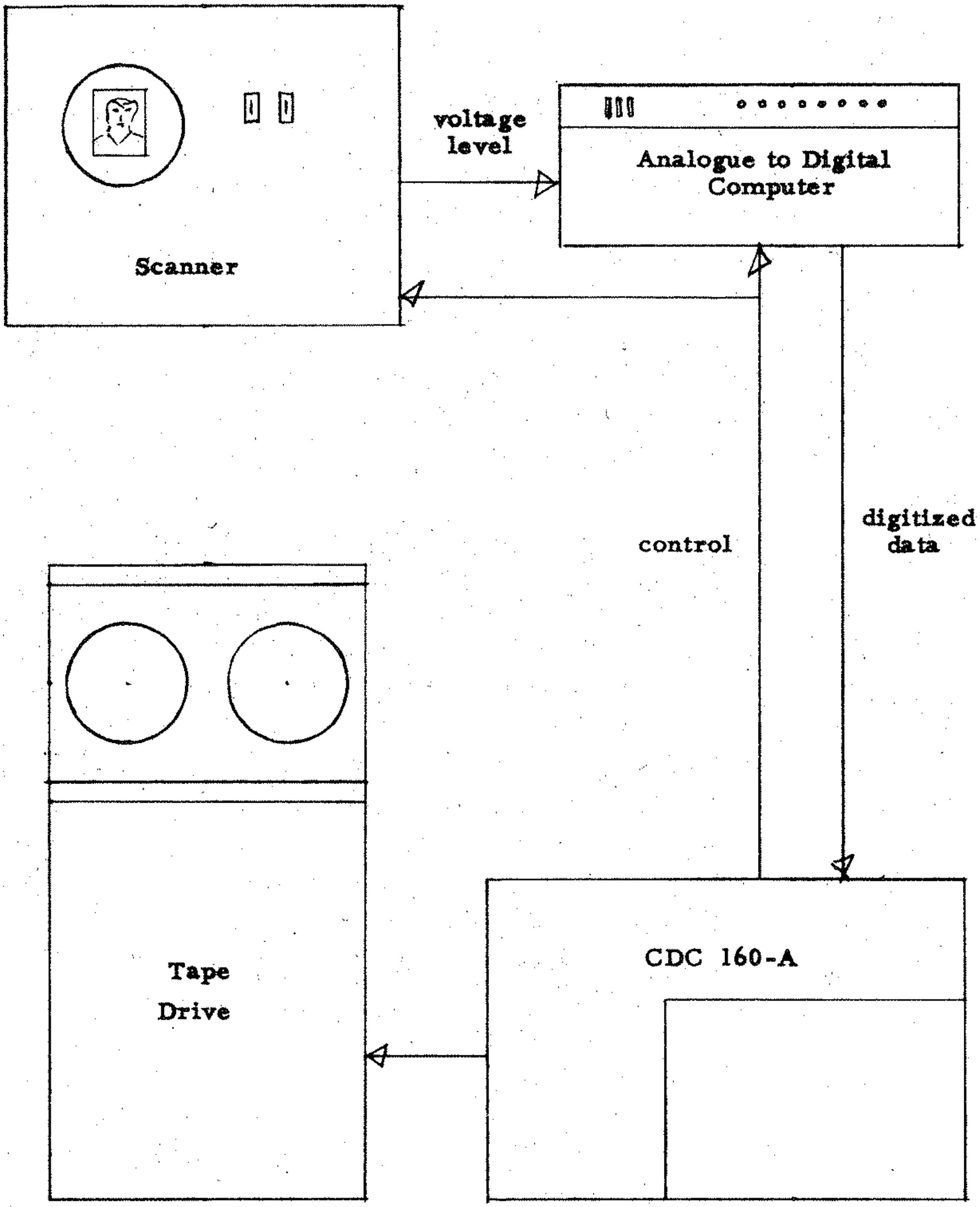


Figure 4

Schematic of New Scanner

the mean is

$$x = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \Big/ (n \cdot m)$$

and the standard deviation is

$$\sigma_x = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - x)^2 \Big/ (n \cdot m - 1)}$$

where n is the number of rows and m is the number of columns. Transform the matrix by

$$x_{ij}' = \frac{x_{ij} - x}{\sigma_x}$$

and the result is a new matrix $[x_{ij}']$ in which the values have been normalized with respect to the average light intensity and the variation in intensity. This would compensate for differences in film density and overall light level.

Variation in the position of the face in the frame can be corrected by translating the face left, right, up, or down until some selected feature (the right eye for example) is in a standard location. The feature can be found manually or by the computer itself.[2] Once it is found the translation is just a matter of shifting the indices on the matrix values, which is carried out by a computer program. In the section on the program NUBLOB there is an example of how this is done.

Other transformations might be useful in certain situations. The whole face could be tilted until some horizontal or vertical requirement were met, such as having both eyes on the same horizontal line. The face could be expanded or contracted until some measurement reached a standard length. For example, the distance between eyes might be made a fixed length, and the rest of the picture scaled to fit that length. Another transformation is rotating the face which is discussed in some detail elsewhere in this report.

---

[2] See Appendix II, "Computational method for determining eye location on a transduced picture," by C. M. Williams.

The scanner itself has certain adjustments which can be calibrated on test cases with the monitor scope. In the first model the overall voltage level and each clip level may be set independently. For the new model, the zero level (no light) and the saturation level (all light) may be set. It is possible to program any transformation on the input values. But for timing reasons, only the simplest tranformations are practical at scan time. An example of what we have done is a scanning program which inverts the incoming values so that a high number represents a dark spot and a low number represents a light spot.

6. Basis of the Abstraction and Recognition Processes.

Consider the form of the picture after it has been scanned and preprocessed. It is an array of light intensity values $\left[x_{ij}\right]$, which look like

$$\begin{array}{cccc} x_{11} & x_{12} & \cdots\cdots & x_{1m} \\ x_{21} & x_{22} & \cdots\cdots & x_{2m} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ x_{n1} & x_{n2} & \cdots\cdots & x_{nm} \end{array}.$$

For those pictures already scanned the values of n and m respectively are 250 and 210. Therefore each picture is represented in the computer by 52,500 light intensity numbers. In general this is far too many to be handled by any recognition program written for a currently available computer. To convert this problem to manageable size the 52,500 values are the input to a program which abstracts a smaller number of values. Our current programs output up to 360 values, which is only an arbitrary number and could be increased.

This abstraction step is crucial. Not only is it necessary for the reason given above, but also we can use this step as an idealized scanner. That is, we can try to select these characteristics which facilitate the recognition process and eliminate those which are unimportant potentially. One of the most fruitful areas of research is the selection of these characteristics. In general the abstraction transformation will be non-linear in nature, so that even with a linear recognition procedure we are not limited to linear decisions. What should the object of the abstraction be? When people talk about recognizing faces, they often refer to specific features, relative proportions of the face, etc. An abstraction scheme based on such behavior is discussed elsewhere in the report. On the other hand, the simplest, fastest method is most desirable from the point of economy. The section on the NUBLOB program details an implementation along these lines.

Schematicly, an abstraction program looks like:

$$\left.\begin{array}{cccc} x_{11} & x_{12} & \cdots\cdots & x_{1m} \\ x_{21} & x_{22} & \cdots\cdots & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ x_{n1} & \cdot & & x_{nm} \end{array}\right\} \text{-Abstraction Program --} \left\{\begin{array}{c} y_1 \\ y_2 \\ y_3 \\ \cdot \\ \cdot \\ y_k \end{array}\right.$$

The set $y_1, y_2, \dots y_k$ will be referred to as a pattern vector. To take a simple example, assume $k=2$ and that y_1 is the heighth of the face and y_2 is the width of the face. Suppose there are photographs of two men, A and B. We measure the heighth and width of each face as it appears in its photograph.

A	y_1(ht)	y_2(width)	B	y_1(ht)	y_2(width)
photo 1	3.2	1.7	photo 5	3.1	2.0
photo 2	3.2	1.6	photo 6	3.2	2.1
photo 3	3.1	1.6	photo 7	2.9	2.0
photo 4	3.2	1.5	photo 8	3.1	1.9

Recording these vectors by their coordinates we get a result presented in Figure 5. The line HP separates the four pattern vectors associated with man A from the four pattern vectors associated with man B. If we then take an unknown picture of one of these two men, take the measurements and record these measurements in the same way, we can hope to recognize the picture by whether the corresponding pattern vector is above or below the line.

Now we have an outline of a recognition program:

(1) All input is reduced to pattern vectors by use of scanner, preprocessing program, and abstraction program.

(2) Known pattern vectors are recorded.

(3) One or more planes are found which devide the space in such a way that no two pattern vectors of different types (persons) fall in the same area.

(4) Unknown pattern vectors are recognized by where they fall with respect to the planes generated in step #3.

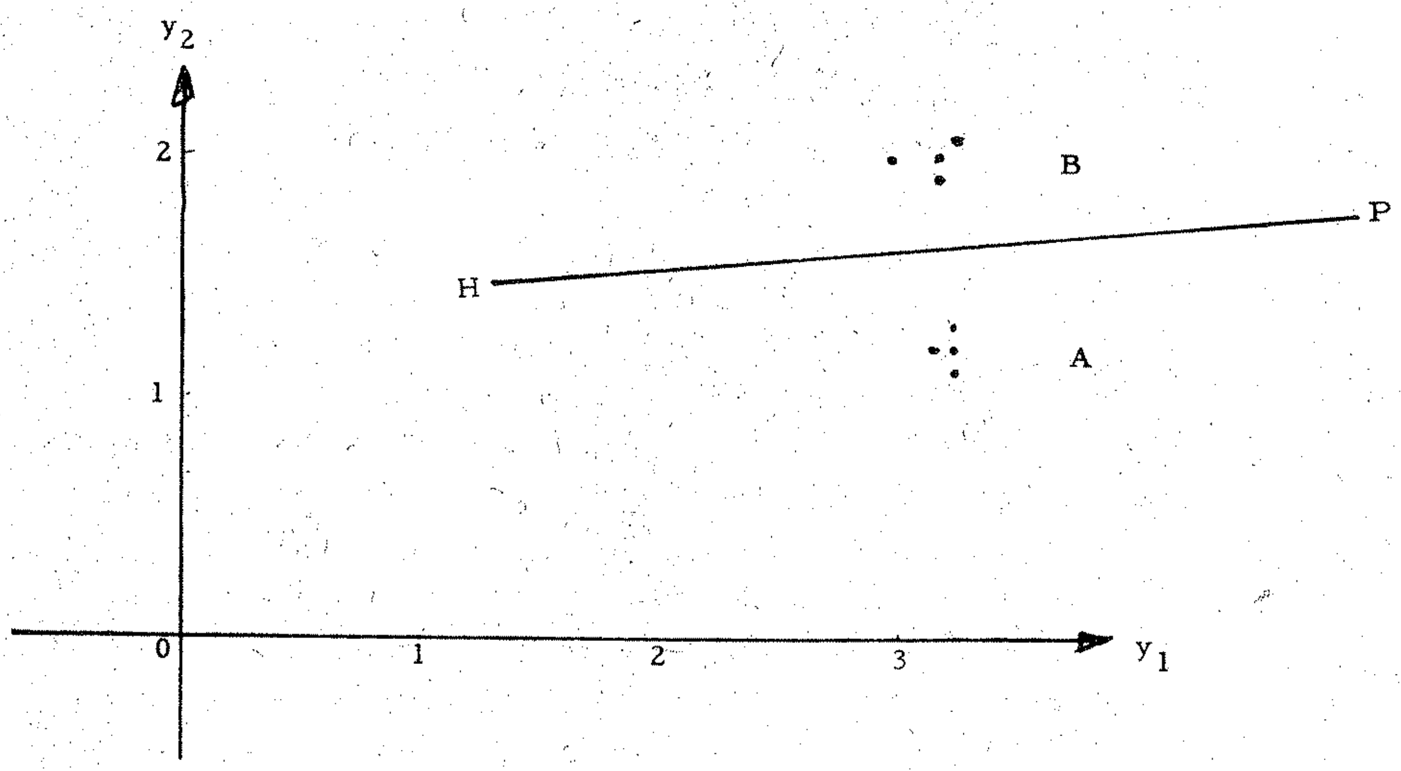


Figure 5

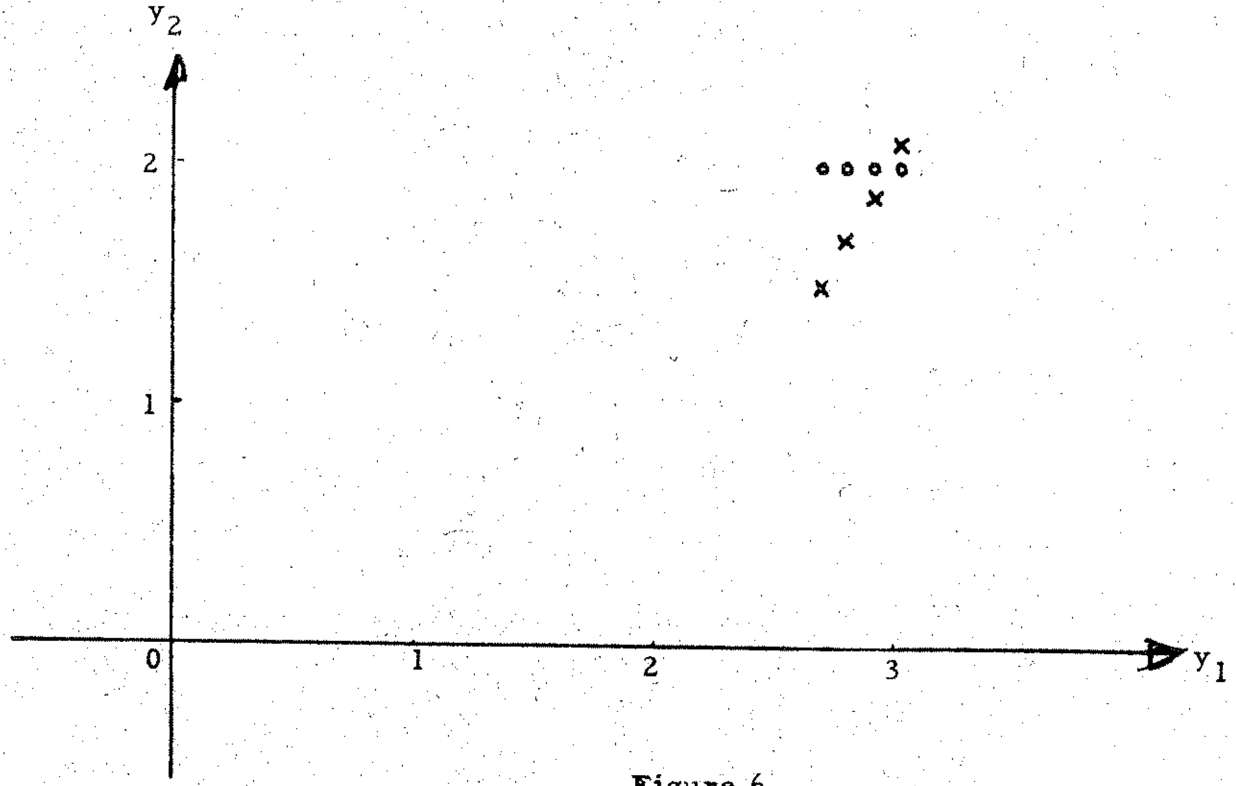


Figure 6

The number of elements, k, in the pattern vector is called the dimension of the pattern space. A two dimensional system was used in the above example because we can draw two dimensional pictures. In practice, the vectors and planes are stored in a computer memory, so that we are in no way limited to a low dimensionality. Using more dimensions cannot hurt the chance of a successful recognition theoretically. That is, if one can distinguish pattern vectors in k dimensions, then he can distinguish those vectors regardless of what additional information is given. But practically, the size, speed, and cost of the computer sets the upper limit on the dimensionality. As available computer memories become larger and the cost per operation comes down, we can economically abstract more dimensions from our data. The original data has 52,500 values per picture, and to date we have abstracted up to 360 dimensions in our experiments. Can we estimate how many dimensions would be needed for a given problem? In the next paragraph we will look at some of the factors involved.

In the worst possible case, where all the pattern vectors are random, Thomas Cover[3] has shown that on the average one needs one dimension for each two training patterns used to find a plane. A good abstraction program of course will not end up with random vectors. To clarify the sort of problem that can come up when a poor choice of abstraction is made or the dimensionality is too small, let us look at another example.

	Man X			Man O	
	y_1	y_2		y_1	y_2
photo 1	2.7	1.5	photo 5	2.7	2.0
photo 2	2.8	1.7	photo 6	2.8	2.0
photo 3	2.9	1.9	photo 7	2.9	2.0
photo 4	3.0	2.1	photo 8	3.0	2.0

---

[3] Thomas M. Cover, "Classification and Generalization Capabilities of Linear Threshold Units," Stanford Research Institute Memo, November, 1963.

Clearly in Figure 6 no line can be drawn which will separate the X's from the O's. In this case, with this choice of dimensions the recognition problem is ambiguous. Suppose we have four different individuals with a number of pictures of each and still only two dimensions. Naming the four people A, B, C, and D the next figure shows a possible configuration. In this example we can separate all four groups one from another with just two planes, HP1 and HP2. (Two is the minimum number of planes that can uniquely separate four groups.) But the planes had to be carefully chosen. If we started by trying to separate the B's and C's from the A's and D's with the first plane, we would have been unsuccessful. This seems trivial in two dimensions, but it represents a definite problem in higher dimensions where one cannot "see" how the vectors are grouped. The next figure shows a case in which the groups can be separated in any fashion.

If we restricted ourselves to two dimensions, we would quickly generate difficult or ambiguous situations as the number of people to recognize grew larger. Exactly the same thing happens at the human level. As long as you know only one red-headed person, he is uniquely identified by that characteristic. But as soon as you meet more than one person with red hair, you must add other characteristics (dimensions) to your identification procedure. For a thorough and technical discussion of linear decision theory in pattern recognition see Highleyman.[4] Many efforts have been made for implementing linear decision techniques with analogue devices, see for example Rosenblatt[5] and Widrow.[6]

---

[4] W. H. Highleyman, "Linear Decision Functions, with Applications to Pattern Recognition," Proc. IRE, June 1962, pp. 1501-1516.

[5] F. Rosenblatt, "The Perceptron -- "A Theory of Statistical Separability in Cognitive Systems," Rept. VG-1196-G-1; Cornell Aero. Lab., January 1958.

[6] B. Widrow and M. E. Hoff, Jr., "Adaptive Switching Circuits," TR No. 1553-1, Stanford Electronics Lab., June 1960.

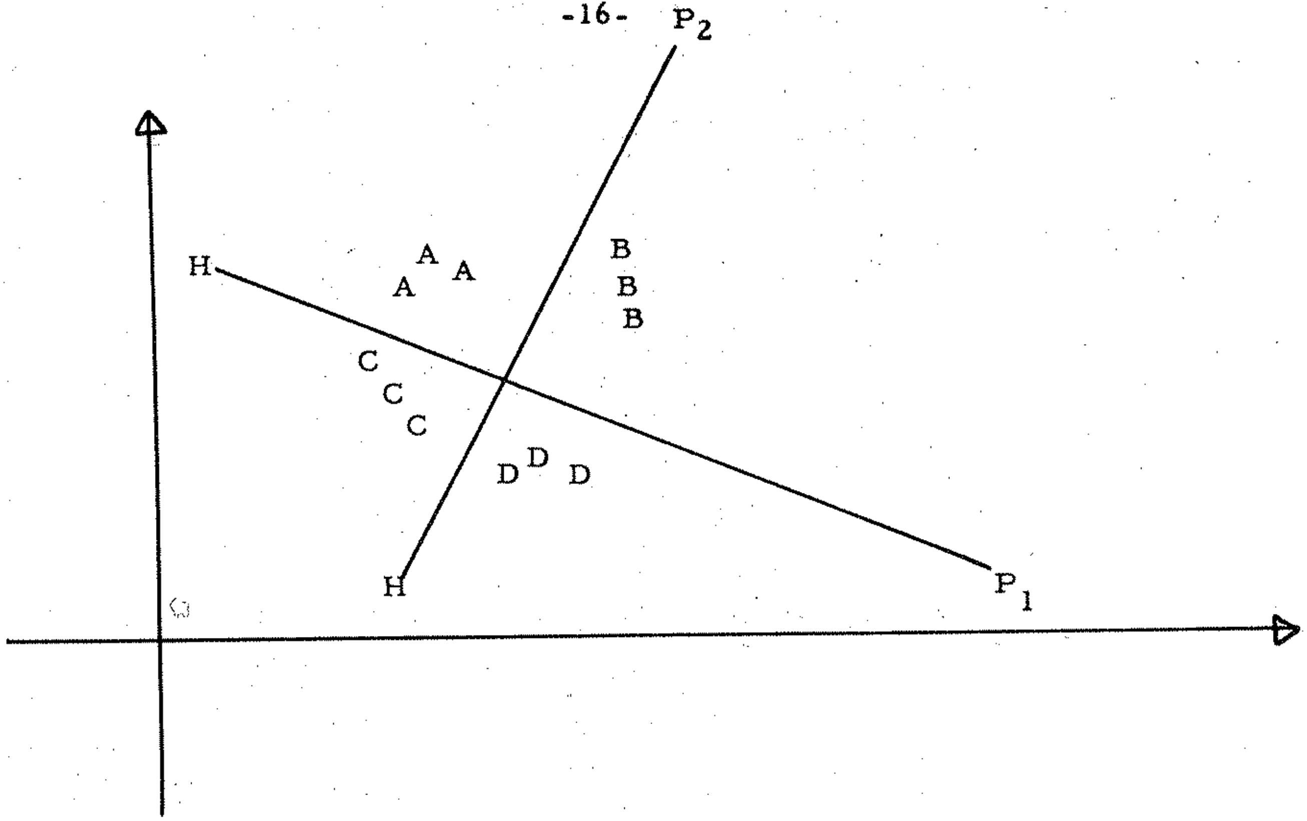


Figure 7

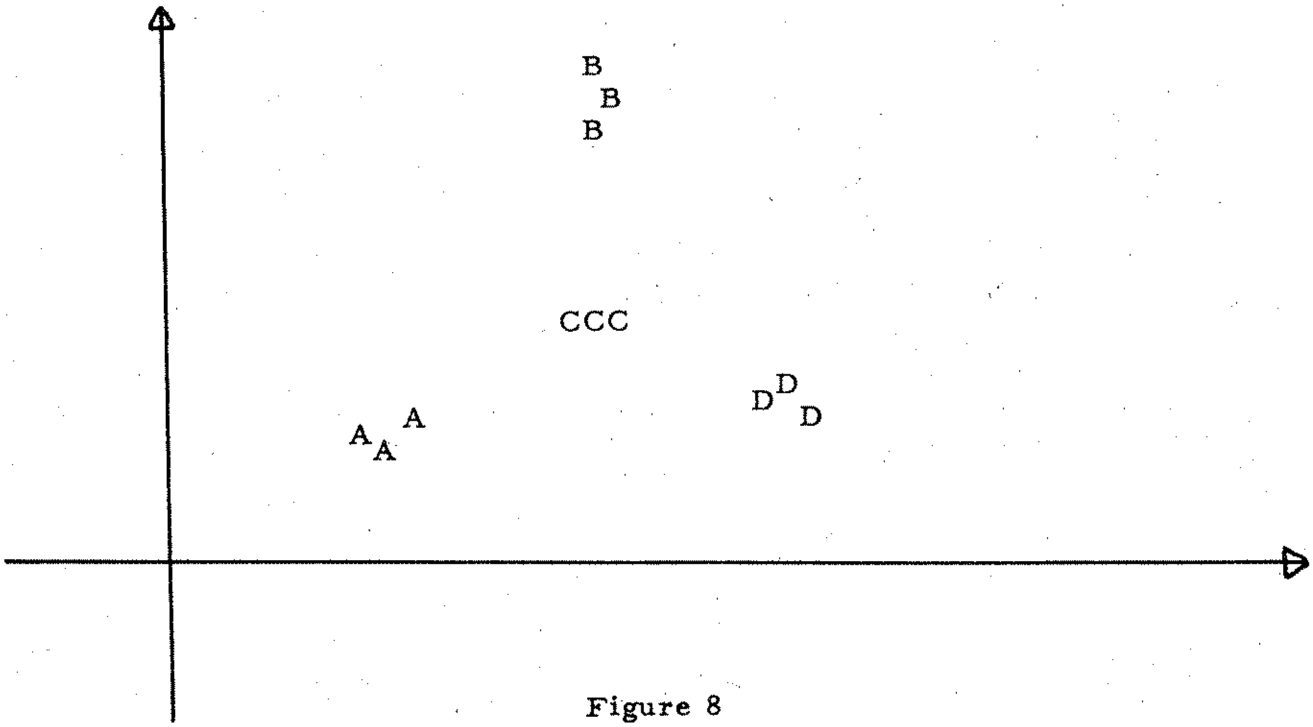


Figure 8

7. An Abstraction Program: NUBLOB

The input to this program is the high density tape format discussed above. It is best thought of as the array of light intensity values. On this matrix we superimpose a sequence of arbitrary rectangular masks. For each mask we sum the point values which it covers. Thus each picture reduces to a set of numbers which are the sums of the values for the light intensities in the respective masks.

In any given run of the program, the length and width of each mask and its relative position to all the other masks is fixed. Any of these variables may be changed from one run to the next. During a run the placement of the set of masks is determined for each picture separately. That is, the masks as a group may be moved up or down or right or left to normalize the face with respect to the masks. (See Fig. 9)

Program Specifications:

1. Input

   A. High Density Tape on Logical Unit #3, 2299 word long records, one picture per record.

   B. Control Card with number of masks, number of pictures to skip on input tape, number of records to skip on output tape. FORMAT (3I4)

   C. One card specifying each mask: left boundary, right boundary, top boundary, bottom boundary, and multiplicative factor. FORMAT (4I4, F10.0)

   D. One card for each picture: horizontal displacement, vertical displacement, number of pictures to skip after this picture (negative number terminates the program after this picture), debug print control number (blank for no debug print out). FORMAT (4I4)

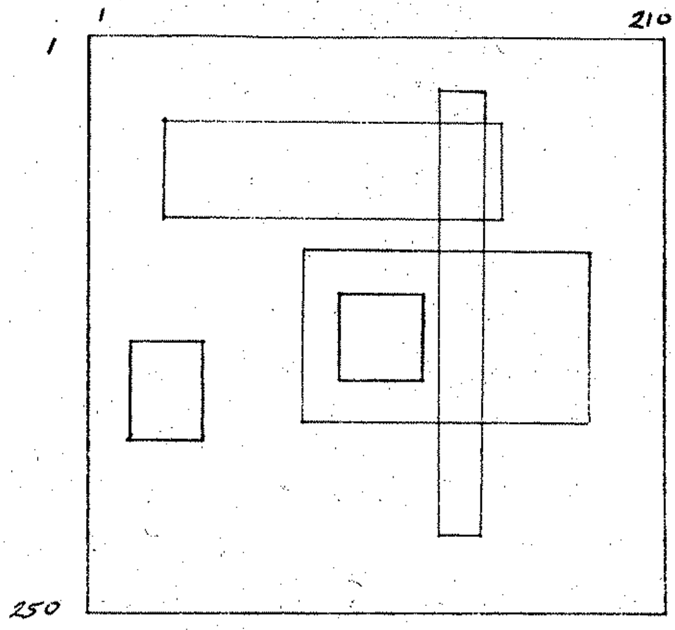
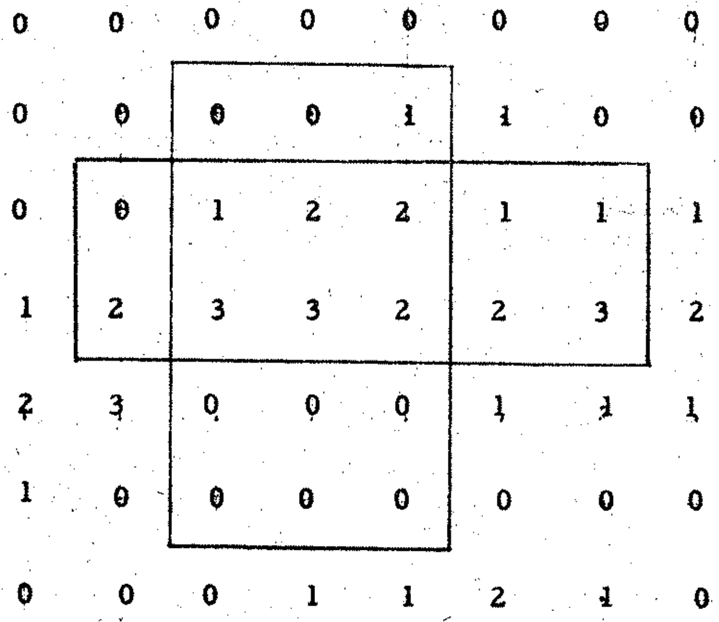


Fig. 9A - Examples of typical masks that can be specified with NUBLOB.



0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0
0	0	1	2	2	1	1	1
1	2	3	3	2	2	3	2
2	3	0	0	0	1	1	1
1	0	0	0	0	0	0	0
0	0	0	1	1	2	1	0

Fig. 9B - Detail of two masks superimposed on an array of light intensity values

Value of wide mask = 22
Value of tall mask = 14

2. Output

   A. High Density Tape on Logical Unit 44, records n+1 words long where n is the number of masks, terminated by an EOF mark.

   B. Printer lists error messages and debug output.

3. Main program written in FORTRAN with one assembly language subroutine called EXPAND which is not part of the FORTRAN library.

4. Timing approximately ten pictures per minute with 100 masks.

8. Another Set of Masks.

In additon to the NUBLOB masking technique described in the last section we used another set of masks which were especially constructed to fit the shape and information content of the face. These were constructed by Iben Browning.

There are a total of 360 of these masks constructed, a few of which are shown in Figure 10.

9. A Recognition Program: BLOBREC

This is a recognition program which accepts input from NUBLOB or any similar abstraction program. One run of this program establishes a plane in the space of the picture vectors. This divides the space into positive and negative halves. For each input picture vector there is an indicator to show if the vector should be on the positive side, the negative side, or is a test vector. The program searches for a (best) plane such that the positive pictures are on the positive side and the negative pictures are on the negative side. Then the test vectors are compared to the plane, and their position is recorded on the output. First the program is trained on pictures which are identified by their indicators; then unidentified picture are read in and recognized by their position with respect to the planes generated in the first stage.

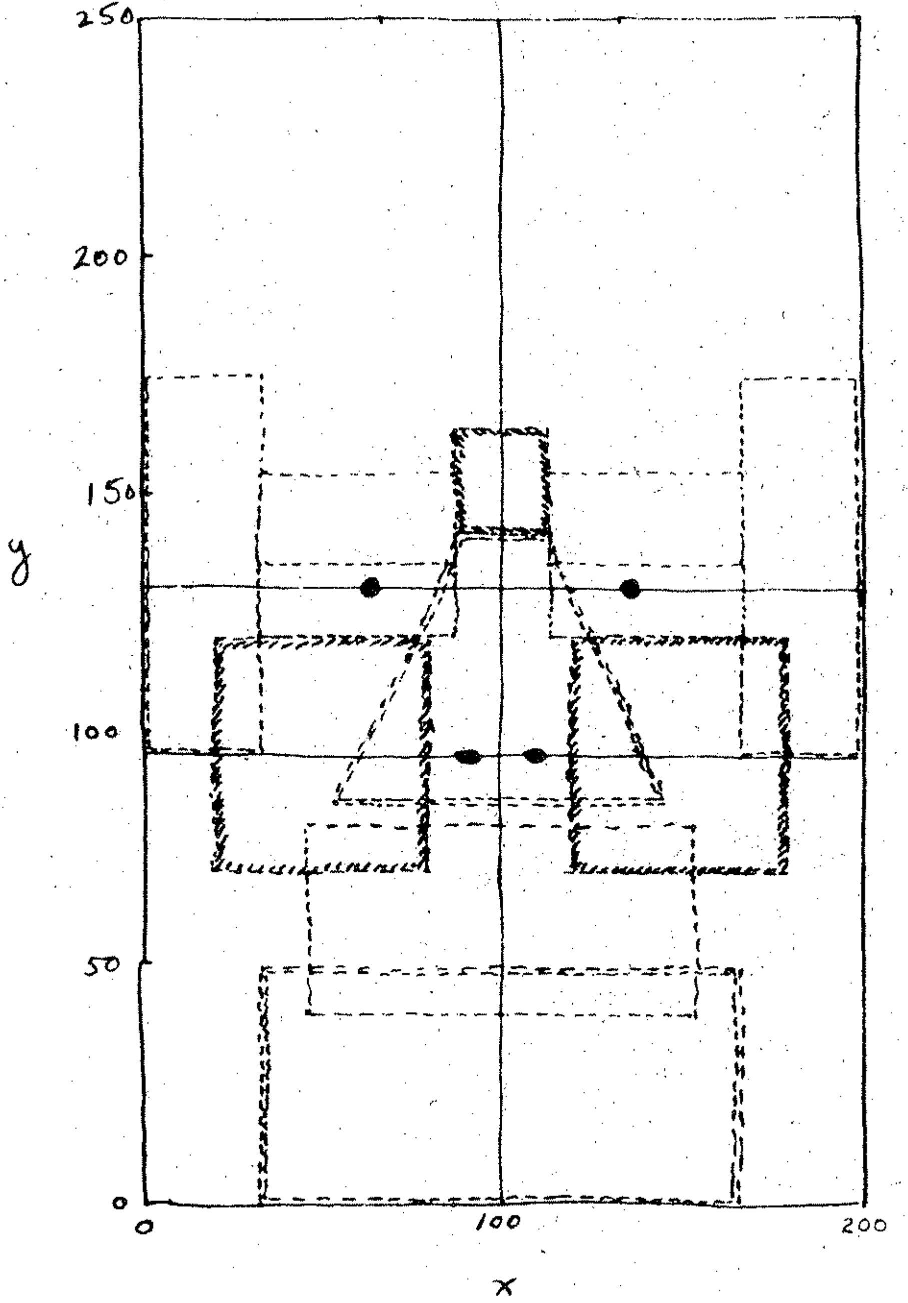


Figure 10

Specially Constructed Masks

After the picture vectors have been read in by the program, the vector are normalized to the unit sphere by dividing each vector by its length. This step is intended to eliminate spurious differences due to over-all light intensity variations in the original photographs. Then all the vectors are imbedded in a space with one more dimension in such a way that the new coordinate value is 1 for each vector. All of the vectors in one of the input groups (the vectors with negative indicators) are reflected through the origin. These operations are not logically necessary; they are essentially bookkeeping tricks so that the running of the program is simplified. The problem of finding a plane which splits the two input groups is now reduced to the problem of finding a homogeneous plane (a plane through the origin) which has all the vectors on one side. As an initial guess, the average of all the vectors is taken as the normal vector of the first plane. From this initial guess the program is iterated to minimize the function $\sum_{i=1}^{\#\text{of vectors}} e^{-sd_i}$, where d_i is the directed distance of the i^{th} vector from the most recent plane and s is variable which is increased when the program is doing well and decreased when the program is having difficulty. s is always greater than 0, so that vectors on the wrong side of the plane have the most effect on the next iteration. And vectors on the right side but very far away have the least effect.

In referring to the best plane, it is with respect to minimizing the function in the last paragraph and limited by the number of iterations the program completes. With this program, two kinds of experiments have been performed. First handwritten letters were used for debugging. These tests were successful, but they are not sufficiently complex to do more than debug the program. More thorough testing, using data obtained from pictures scanned by the machinery described above, showed promising signs at the time of termination of this contract. Further work is needed.

Program Description:

1. Input

   A. Magnetic tape with picture vectors.

   B. Card with number of vectors, number of dimensions, number of vector components to be eliminated, number of iterations, number of subiterations, number of s's, length of picture vectors.

   C. Card with step size, decrement in step size, initial s, step control variable for s, increment in step size.

   D. Card with list of indicators for input vectors.

   E. Card with list of components to be removed from input vectors.

2. Output

   A. Final output is a listing of each vector and its directed distance from the plane and the coefficients of the plane.

   B. Intermediate output can include the above plus change in s and step size and a list of the vectors giving the greatest difficulty.

3. The main program and its subroutines are written in FORTRAN by Charles Williams.

4. Timing data is incomplete, but for a sample run with 20 vectors of 100 components each, the time for one iteration was less than one minute.

6 Mar 1964

APPENDIX II

COMPUTATIONAL METHOD FOR DETERMINING EYE LOCATION ON A TRANSDUCED PICTURE

Charles M. Williams

The most important single process in facial recognition is the initial determination of the exact location of the face. Unless this is done accurately, measurements performed on different pictures cannot be meaningfully compared. The most obvious feature to use in determining facial location appears to be the eye, more specifically, the eye ball. Consequently the supposed location of the center of the right eye was initially given to the recognition programs, and all measurements were made relative to this.

Unfortunately the determination of the value of this parameter by humans was laborious and not very accurate. In addition, the process requires a completely separate computer run to print out the pictures. The main source of inaccuracy was that the shadowing about the eye varies considerably with the direction of the lighting. When the right eye lay in the shadow of the nose its general location was hazy and criteria for judging its center was inexact at best. It was hoped that the computer could be used to ease the task.

In the ensuing computer investigations an equal number of pictures of two subjects was used. The pose was head-on with the direction of lighting being varied in roughly equal increments from one side to the other. The light intensity was constant for all the pictures.

In the following discussion a mask is defined to mean a normalized $r \cdot m$ dimensional vector obtained from points selected from an $r \times m$ rectangular area (See Figure 1) on the original scanned picture. The location of a mask is defined to be the center of its rectangular area.

An 11 x 11 rectangle is large enough to just cover the eye ball, for the scanning density used.

The initial experiment was to compute an average eye from the normalized vector sum of 11 x 11 masks located at the centers of the right eyes of several pictures. This average eye was then compared by means of dot product correlations with all masks in an 11 x 11 area of each of 20 different right eyes. Although the range of values was not large, the highest correlation for each picture came from the mask centered at the eye.

An accidental clerical error yielded an interesting result. The high correlation on one of the pictures used in determining the average eye indicated an eye location which did not agree with the original input. A check revealed that the high correlation was correct. This indicates that it may be possible to finely focus the average eye obtained from rough data by iteratively passing the computer found center masks back through the averaging process.

Vertical Distance from Eye

3	.81	.79	.80	.73	.75	.58	.63
2	.83	.75	.80	.81	.80	.68	.71
1	.70	.77	.76	.86	.65	.70	.72
0	.73	.86	.80	.89	.81	.70	.73
-1	.65	.76	.80	.81	.76	.77	.64
-2	.70	.72	.72	.70	.71	.65	.67
-3	.71	.70	.77	.69	.70	.67	.71
	-3	-2	-1	0	1	2	3

Horizontal Distance from Eye

Typical correlations in the neighborhood of an eye. The center is circled.

Unfortunately the structure of the correlations about the highest value indicate that gradient techniques would not be reliable for finding it, and searching the entire picture would be prohibitively slow. It was hoped that hyperplane methods might be used to get close to the eye faster. The idea was that one hyperplane would be used to determine whether a given mask was left or right of the eye center while another would test whether it was above or below. A unit step taken in the direction indicated would yield the center of another mask which would be processed in like manner. Hopefully this method would result in a bee line path to the center of the eye. Exhaustive searching near the center using correlations or some other method would yield the true position. Investigation was restricted to the left or right case.

Using 7 randomly chosen 11 x 11 dimensional masks from each of the 16 pictures I obtained a hyperplane which yielded excellent results on all the pictures as long as the eye center was present in the mask tested. This method appears, therefore, to be quite workable for moderately close in searching.

Vertical Distance From Eye

4	-12	-9	-4	3	8	11	15	17	17
3	-18	-15	-9	-1	7	12	15	18	19
2	-22	-21	-13	-3	6	12	17	19	20
1	-22	-21	-14	-3	7	14	19	22	22
0	-22	-22	-15	-2	7	14	20	23	24
-1	-21	-21	-16	-3	8	15	21	24	24
-2	-20	-20	-14	-2	9	16	23	25	24
-3	-13	-14	-10	1	12	19	24	25	23
-4	-9	-11	-7	2	12	19	24	24	23
	-4	-3	-2	-1	0	1	2	3	4

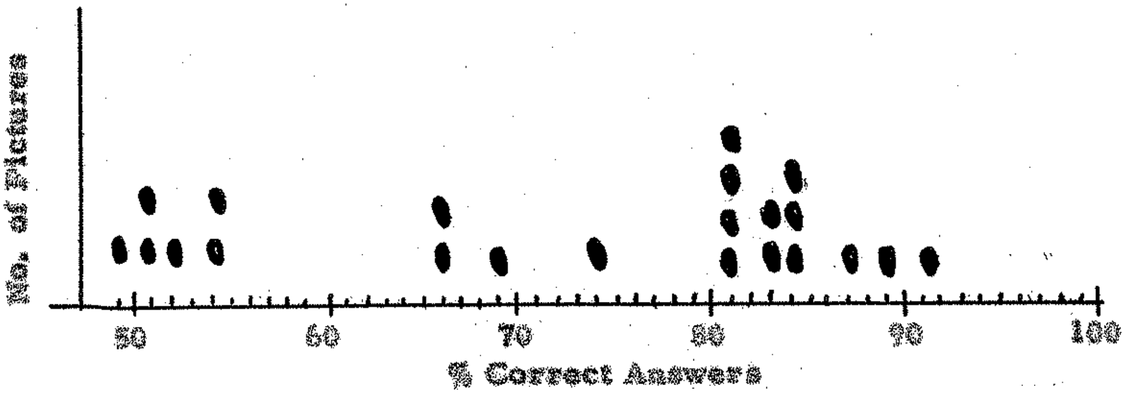
Horizontal Distance from Eye

Hyperplane scores in the neighborhood of an eye center. Note that scores are negative to the left of center and positive to the right.

In extending this idea to finding the eye from greater distances it is evident that merely enlarging the masks has the effect of increasing their dimension by the square of the distance. Therefore small masks were formed by randomly choosing points from large rectangular facial areas. The arrangement of points within a rectangle was constant for all masks.

Using 100 dimensional masks over 21 x 21 rectangular areas with 7 randomly chosen masks learned from half of the pictures, I obtained fair success. Using the criterian that a hyperplane score was correct if it had the correct sign, I checked every other mask in a 21 x 21 rectangular area about the eye center in each picture omitting those masks on the vertical line through the center. Of 22 pictures evaluated in this way, 12 scored over 80% correct answers and only 1 scored under 50%. Interestingly enough this worst picture at 49% was one of those whose sample masks were used in the learning process. Four of the sample masks from it, however, were not split properly by the hyperplane.

Examination of the worst scores showed they came from unshadowed eyes, indicating that probably great improvement would result if shadowed and unshadowed eyes were treated separately. Secondly there is conceivably an optimum arrangement of the mask points in the rectangles which can probably be found only by trial and error.



In conclusion the hyperplane method for finding the eye appears to work quite well. In fact even in the worst cases (with the exception of one) results were better than chance indicates that the method may be improved to yield consistantly good answers. In addition the method appears to be nicely adapted for determining the general location of other facial features as well. Conceivably it may have a value in determining such parameters as facial rotation and tilting. The main benefit of course is that the method is completely general and is not dependent upon pictures being transduced into a form that people can readily recognize.

10	-10	-10	-9	-1	8	19	2	8	6	9	20
8	-7	-2	-4	-6	-5	10	3	16	22	19	36
6	-6	-8	-1	-5	-2	3	-1	12	16	23	27
4	-4	-7	-10	-15	-16	1	7	17	13	11	18
2	-5	-11	-11	-14	-14	4	9	17	15	13	18
0	-5	-10	-10	-12	-2	8	3	10	16	24	23
-2	-4	-7	-5	-8	-11	9	6	13	10	11	14
-4	-11	-20	-17	-18	-10	10	3	9	4	2	-1
-6	-14	-21	-18	-10	-0	9	-1	7	1	-3	-3
-8	-21	-23	-10	-3	1	6	-4	3	-6	-5	-1
-10	-18	-21	-10	2	5	4	-4	-4	3	-2	1
	-10	-8	-6	-4	-2	0	2	4	6	8	10

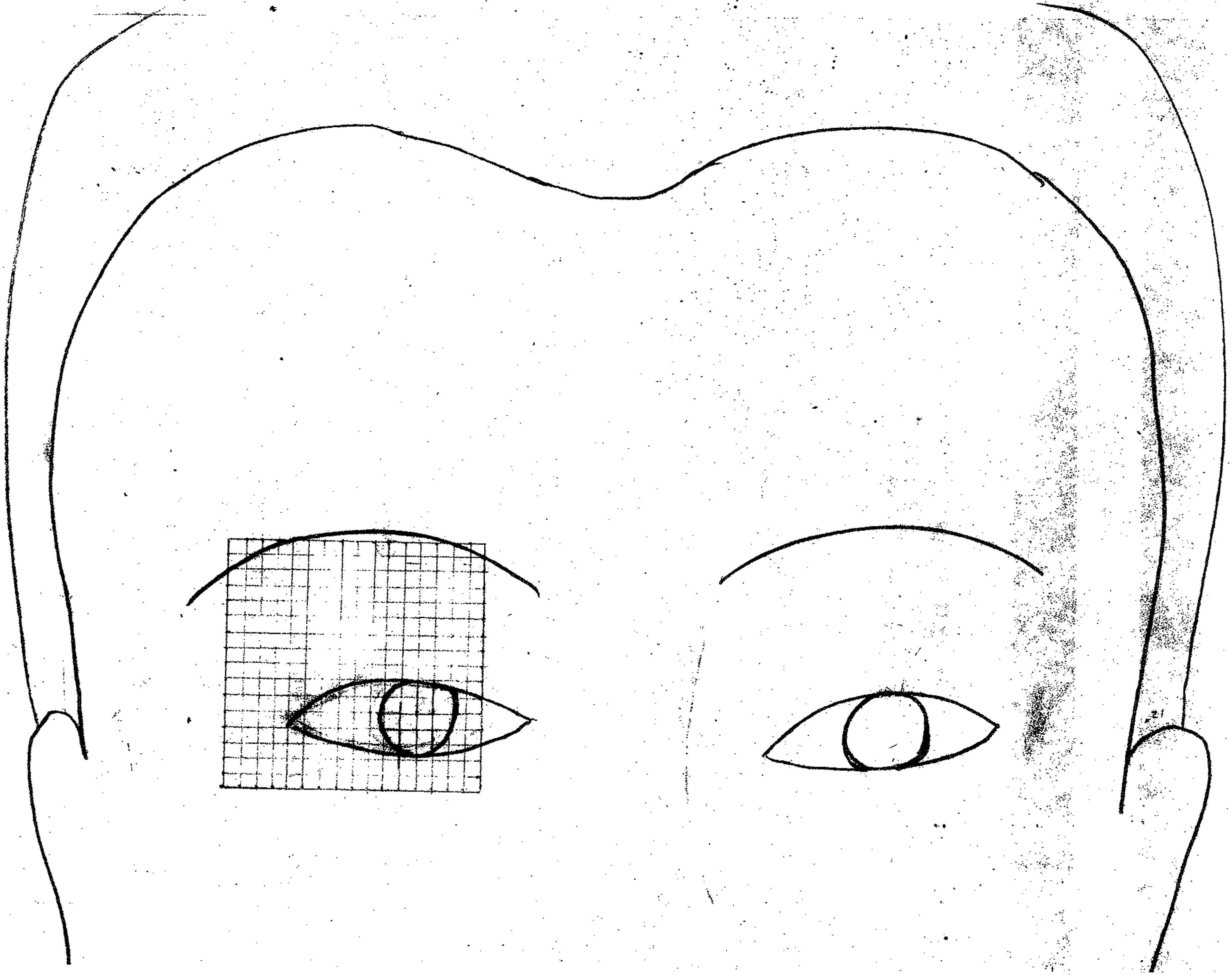
Horizontal Distance From Eye

86% Correct: Hyperplane scores for unlearned eye.

10	7	11	15	22	26	22	7	6	8	17	-6
8	6	2	18	27	14	5	4	8	12	20	1
6	0	0	12	5	1	-8	1	2	2	22	11
4	-1	4	8	-10	-10	7	1	5	12	20	5
2	-5	-6	-5	-8	-10	-9	-4	-5	9	19	6
0	-16	-16	-9	-7	-1	3	2	8	13	18	11
-2	-18	-14	-18	-22	-16	-6	6	6	9	12	7
-4	-24	-22	-29	-20	-17	-4	-3	6	11	16	7
-6	-23	-17	-26	-15	-14	1	6	7	9	13	7
-8	-25	-18	-23	-14	-7	-4	4	10	7	4	1
-10	-21	-10	-7	-12	-1	6	2	5	5	8	1
	-10	-8	-6	-4	-2	0	2	4	6	8	10

Horizontal Distance from Eye

81% Correct: Hyperplane scores for unlearned eye.



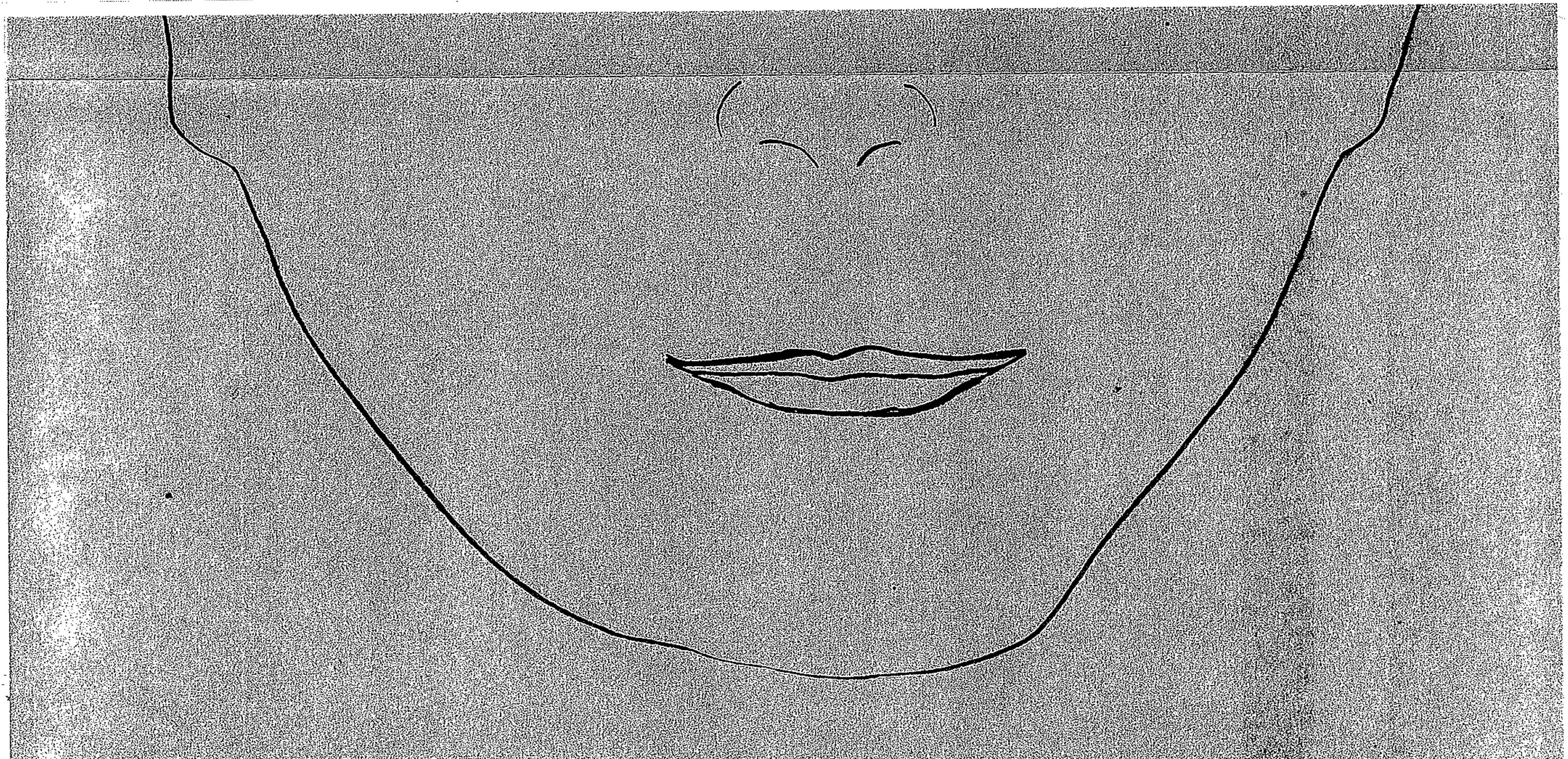


Figure 1

Rectangular Mask for Locating Eye

APPENDIX III

RECONSTRUCTED PICTURES

The information on the photographs which is used in the recognition process, is fed into the computer by an optical scanner. This process is described in Appendix I.

In order for humans to see what picture the computer has in its memory it is desirable to have the computer reconstitute the picture and print it out. One such reconstructed picture is shown in Figure 1. In this example the scanner measured only four levels of light intensity and, consequently the quality of the reconstituted picture is quite low. This quality can be improved routinely by increasing the number of levels.

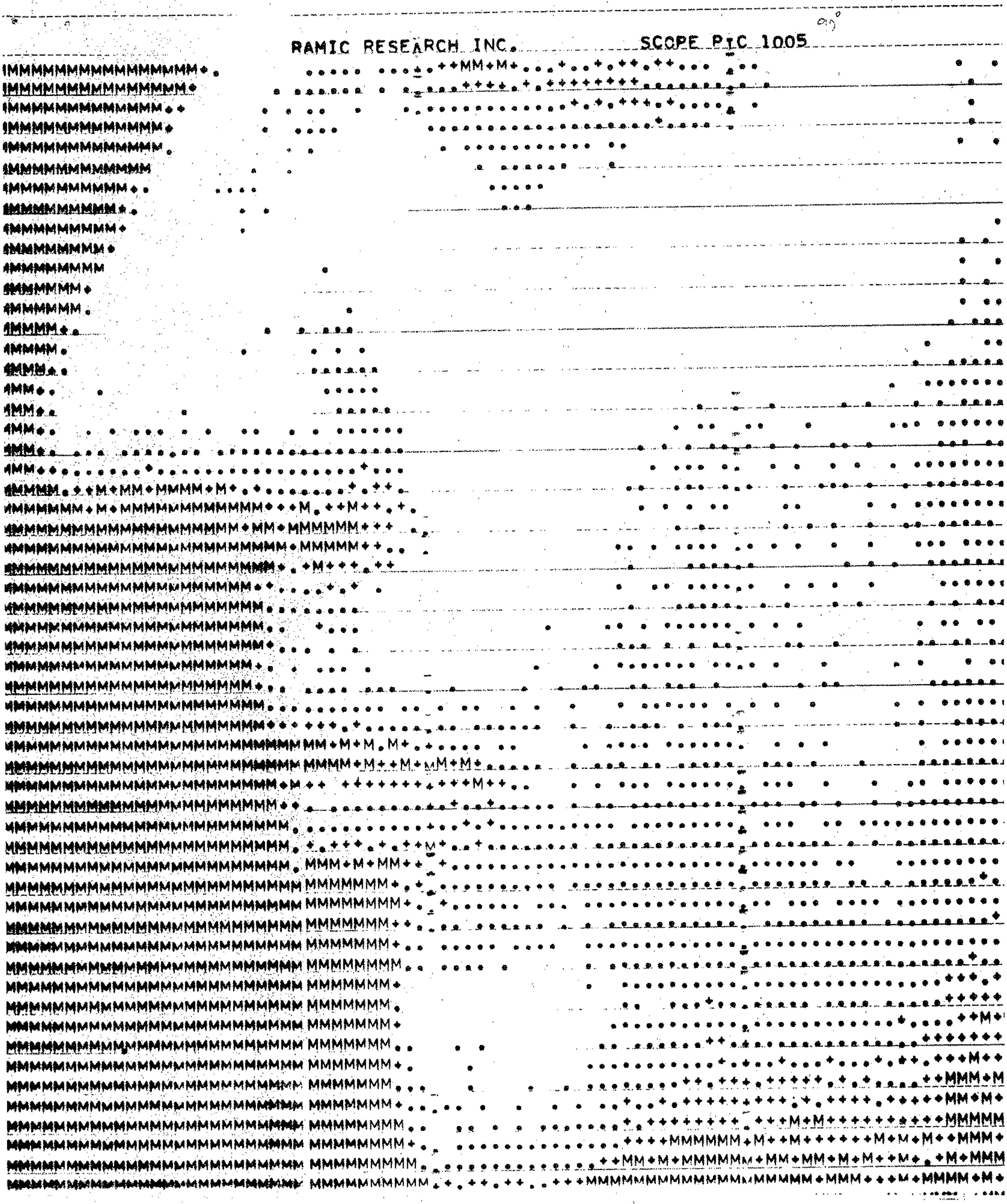


Figure 1

A Reconstructed Picture with 4 Levels of Light Intensity

6 Nov 1964

APPENDIX IV

ROTATION OF FACES

Helen Chan, Lawrence Cowdery, and W. W. Bledsoe

NOTATION

Suppose we have two views of the same face where one is a frontal view and the other is a horizontal rotation of it through an angle θ (See Figure 1). We will establish a coordinate system on the frontal view. At each point (x, z) on this frontal view, we will let y represent the elevation of the face there. This elevation is measured from some fixed plane below the surface of the face (See Figures 2 and 3). It is this elevation, y, that we first wish to determine for each point (x, z).

To each point (x, z) on the frontal view there is a corresponding point (x', z') on the rotated view, where x' is given by the simple rotation equation.

$$(1) \qquad x' = x \cos \theta + y \sin \theta,$$

$$z' = z.$$

It should be noted that if the rotation is not horizontal (i.e., the head is also tilted) then the expressions for x' and z' will be more complicated. We will restrict out discussion here to the case of horizontal rotations but more general rotations can be handled in a similar way.

Equation (1) can be used to determine y when x and x' are known. Thus

$$(2) \qquad y = (x' - x \cos \theta) / \sin \theta.$$

THREE DIMENSIONAL STRUCTURE OF A FACE.

The three dimensional structure of a face can be determined as follows. First two photographs of the same face are taken, one frontal

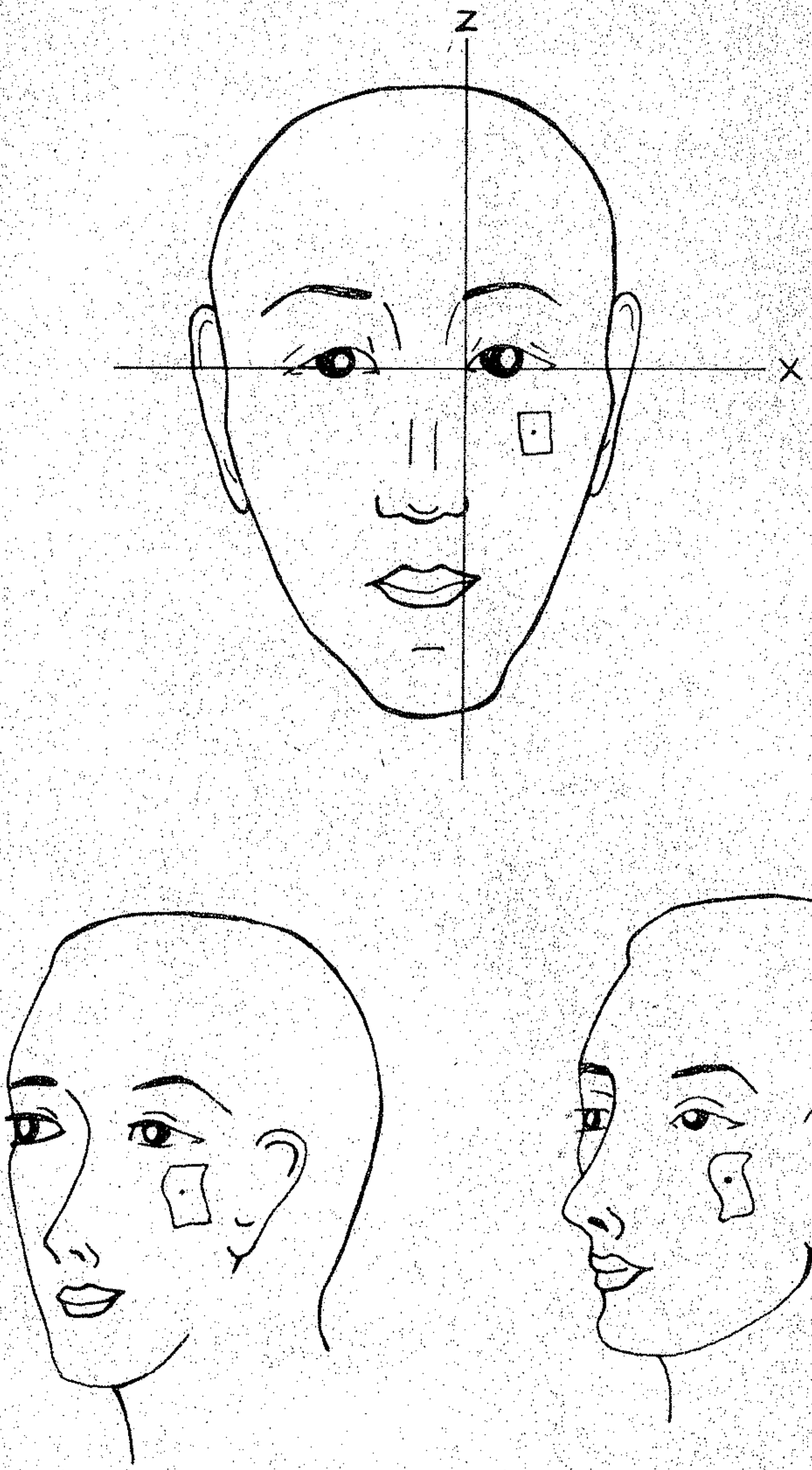


Figure 1

Frontal and Rotated Views

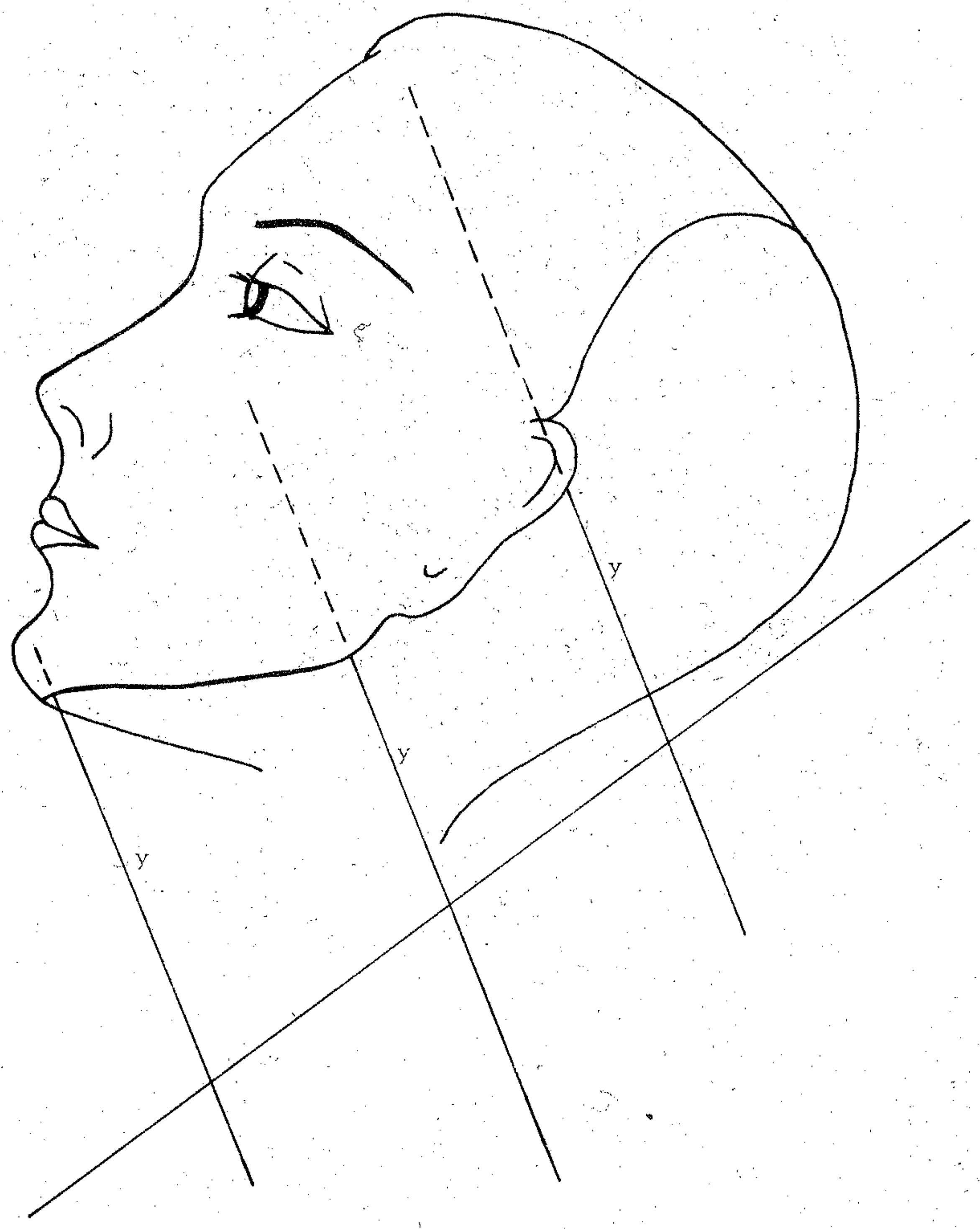


Figure 2

Elevations, y, to Some Points on the Face

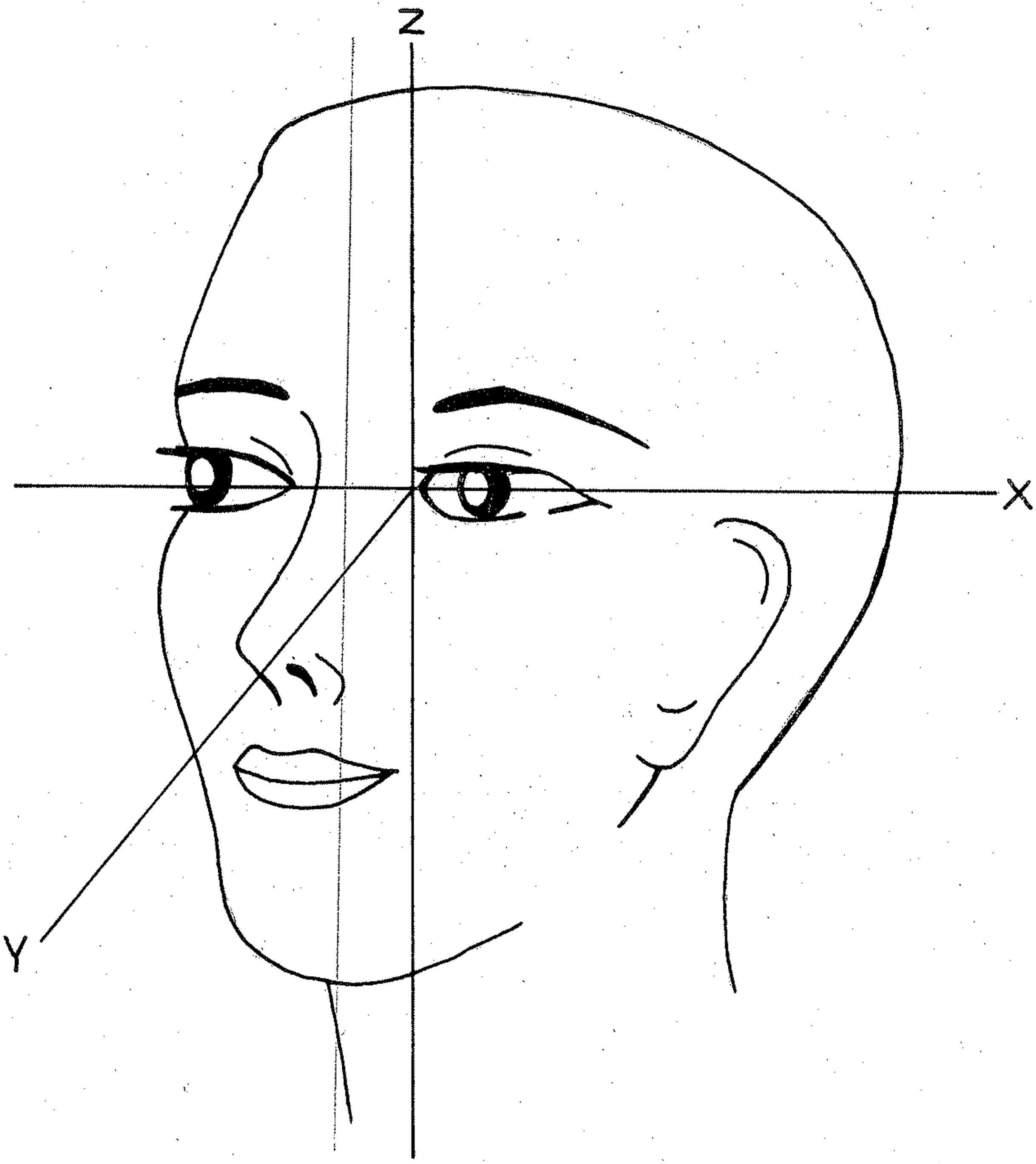


Figure 3

Three-Dimensional Coordinates on Face

and one rotated horizontally through an angle θ (See Figures 1, 4 and 5). Artificial marking are put on the face before the photographs are taken in order to make possible the measurements x and x′ on these photographs. (See Figures 4 and 5) Equation (2) is then used to determine the elevation y at each of the points marked. The elevations at other points are determined by interpollation. In all of the cases considered, elevations are computed and rotations are made for only one side of the face, the side facing the camera.

The three dimensional structure was determined for the face shown in Figure 4. These results are given in Table 1 and were used to produce other rotations on this face. A rotation angle of $\theta = 45°$ was used for this particular case to determine the three-dimensional structure.

ROTATING A PICTURE TO THE FRONTAL VIEW

Suppose we have another view of our subject with rotation angle ϕ and wish to transform it into a frontal view. This can be accomplished as follows by the use of the rotation equation (1).

Suppose that the light intensity on the rotated face is given by the intensity function $I'(x', z')$. (I' is just degree of whiteness measured at each point by the scanner). Then we will "build" a new frontal view of this face by assigning an intensity function $I(x, z)$ of it, where

$$I(x, z) = I'(x', z'),$$

$$x' = x \cos \phi + y \sin \phi, \qquad (3)$$

$$z' = z.$$

It should be noted that the rotation angle, ϕ, and the elevations, y, of the frontal view were required for this process. We will say that

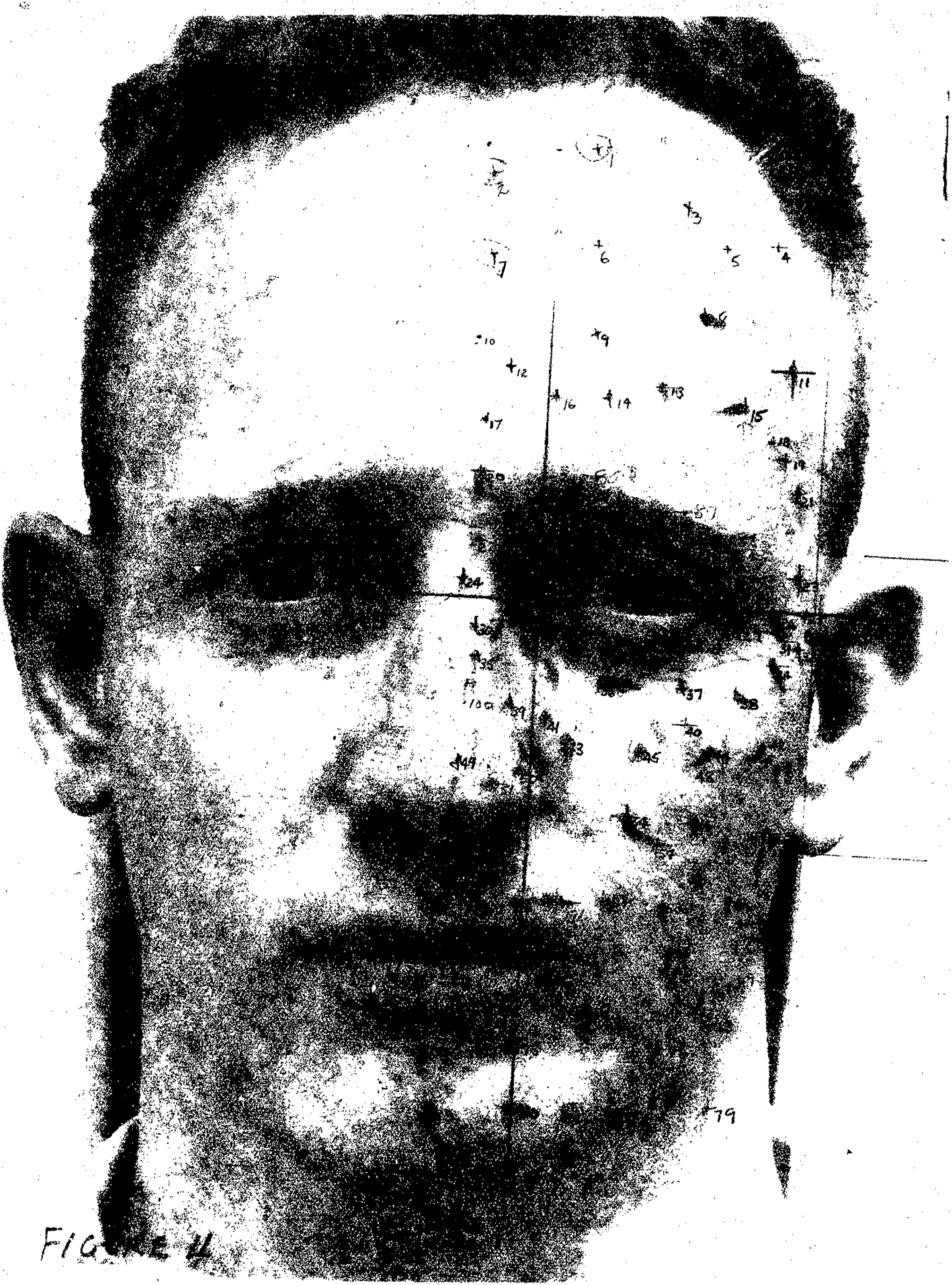


Figure 4

Face with Markings for Determining 3-Dimensional Structure

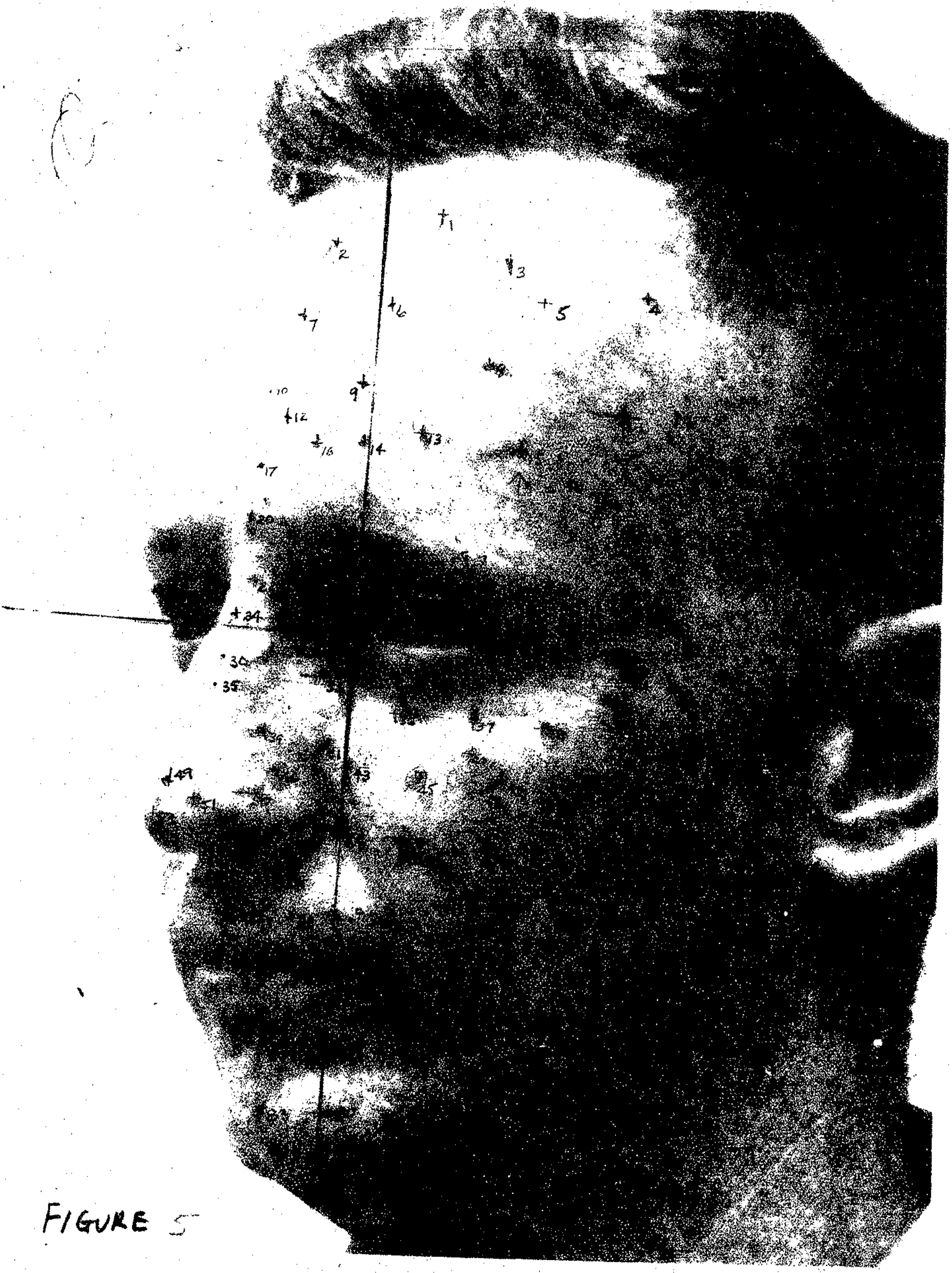


FIGURE 5

Figure 5

Face to be Rotated

Z (INCHES)	X (INCHES)				
	-.70	-.55	-.40	-.25	-.10
3.37	-.0060	.0487	.1035	.1583	.2287
3.22	-.0930	-.0383	.0165	.0713	.1171
3.07	-.1800	-.1253	-.0705	-.0157	.0055
2.92	-.0334	-.2018	-.1700	-.1381	-.1062
2.77	-.1126	-.2185	-.2559	-.2240	-.1921
2.62	-.1916	-.2977	-.4035	-.3099	-.2780
2.47	-.2711	-.3769	-.4028	-.3701	-.3497
2.32	-.5356	-.4984	-.4612	-.4240	-.4212
2.17	-.5910	-.5538	-.5166	-.4793	-.4962
2.02	-.6464	-.6092	-.5719	-.5656	-.5377
1.87	-.6652	-.6488	-.6323	-.6060	-.5792
1.72	-.6962	-.6798	-.6633	-.6464	-.6277
1.57	-.7272	-.7108	-.6946	-.6762	-.6578
1.42	-.7815	-.7515	-.7215	-.7064	-.6879
1.27	-.8066	-.7766	-.7465	-.7307	-.7105
1.12	-4.1833	-.8535	-.7246	-.5958	-.7359
.97	-4.1516	-1.9226	-.7471	-.6182	-.7612
.82	-.8683	-.8018	-.7353	-.6688	-.6806
.67	-1.1331	-.7690	-.7025	-.7053	-.5429
.52	-1.1004	-.6982	-.6601	-.4977	-.3354
.37	-.6746	-.7989	-.5514	-.2902	-.2858
.22	-.7428	-.7923	-.5241	-.3115	-.0944
.07	-1.1242	-.9005	-.6768	-.4924	-.1784
-.08	-.8563	-1.0351	-.8274	-.6069	-.2499
-.23	-1.3849	-1.1507	-.9165	-.5822	-.3133
-.38	-1.1250	-1.1047	-1.0055	-.6682	-.3959
-.53	-1.1962	-1.1759	-1.0451	-.7872	-.4984
-.68	-1.3396	-1.2921	-1.0760	-.8793	-.6193
-.83	-1.3987	-1.3347	-1.1211	-.8895	-.6612
-.98	-1.0171	-1.3778	-1.1571	-1.5996	-.5751
-1.13	-1.1062	-1.4389	-1.2580	-1.0442	-.8874
-1.28	-1.5145	-1.4753	-1.4358	-1.3230	-1.1662
-1.43	-1.5512	-1.5117	-1.3216	-1.0009	-.7593
-1.58	-1.5937	-1.2336	-.9635	-.9144	-.3203
-1.73	-1.3506	-1.0262	-.5446	-.4342	-.3133
-1.88	-1.1432	-.6431	-.5328	-.4224	.3301
-2.03	-.8416	-.5855	-.5210	-1.2586	-.2000
-2.18	-.5670	-.4540	-.4836	-.5166	-.4219
-2.33	-.4952	-.4666	-.4041	-.4618	-.2765
-2.48	-.4234	-.3948	-.3382	-.2948	-.1516
-2.63	-.3652	-.1279	.1094	-.0953	-.2381
-2.78	-.2643	-.4980	-.7316	-.1798	-.0910
-2.93	-.1942	-.4279	-.6616	-.1033	-.0501
-3.08	-.1242	-.3579	-.0530	-.0139	.0806
-3.23	-.0340	-.4587	-.0047	.0360	.0671
-3.38	.0513	-.3733	.0031	.0064	-.0205
-3.53	.0367	.0524	-.0323	.0142	-.0501

Table I

Elevations, y, at various points on the face shown in Figure 4.

equations (3) rotate the face through an angle ϕ. Actually what really happens is a mathematical rotation, or more accurately, a mathematical reconstruction of the face in its rotated position.

This procedure was tried on a pose of the face shown in Figure 5 where the angle of rotation is 45°. This face was rotated from 45° through 30° to a final 15°. The original and "rotated" faces are shown in Figure 6.

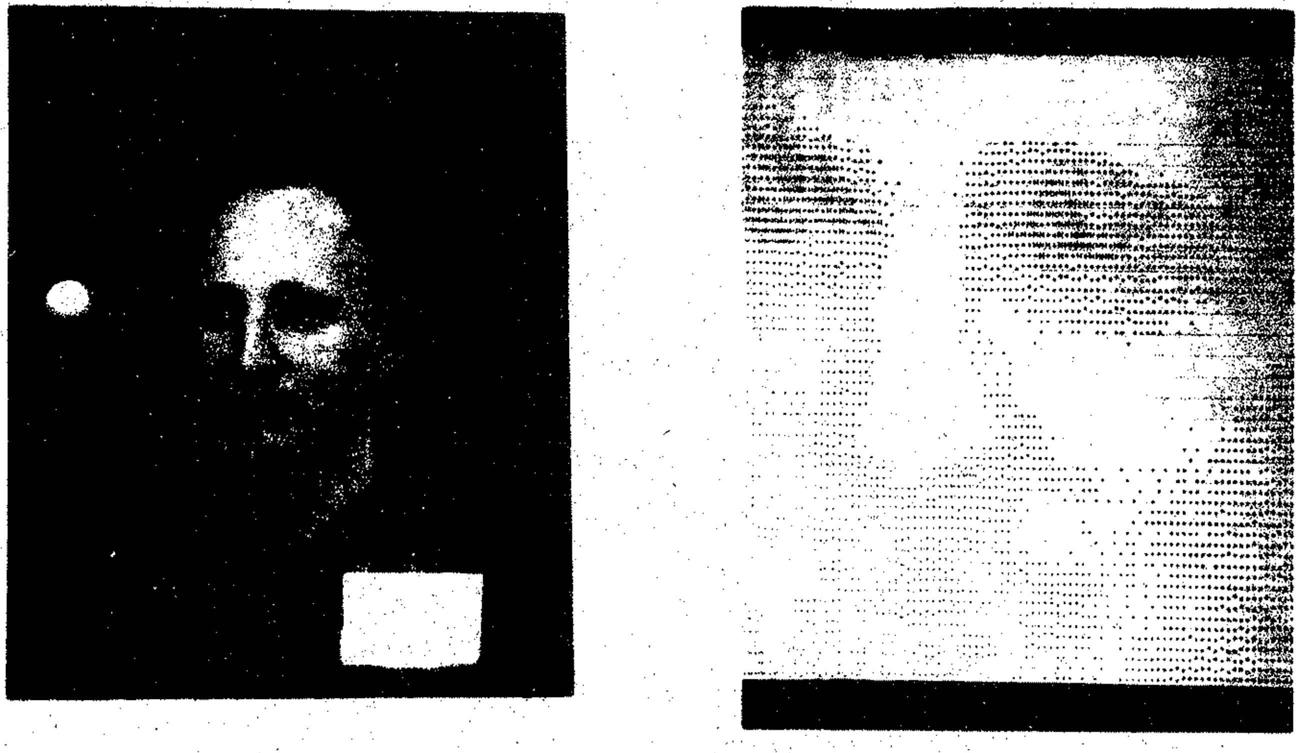
The scanner that was used to transduce these photographs gave only four levels of light intensity; consequently, the quality of the reconstituted pictures is quite poor. However, they are good enough to indicate the feasibility of this type of mathematical rotation. We feel that the quality of this output could have been greatly improved if the 3-dimensional structure of the face, had been more precisely determined. This could be obtained by the use of more markings on the face or, perhaps better, the use of a shadow technique like that described by Hayward and Hilton.[1] Also the utilization of the new scanner described in Appendix which produces 64 levels of intensity (instead of 4) should greatly improve the quality of this output. Also in Figure 6 is shown a photograph corresponding to the reconstructed picture.

In these examples we are able to rotate the faces of Figure 6 only if we know precisely the angle of rotation ϕ, and the three-dimensional structure (the elevation values, y) of the face. Normally, these would not be known for a new photograph that one is asked to identify. Thus, in order to rotate a new photo to the frontal position we must somehow obtain:

1. The rotation angle ϕ, and
2. The elevations y.

---

[1] H. L. Hayward and A. M. Hilton, "Pattern Recognition for Diagnosis of Facial Characteristics," Data Processing for Science/Engineering, Sept. 1963, pp. 31-34.

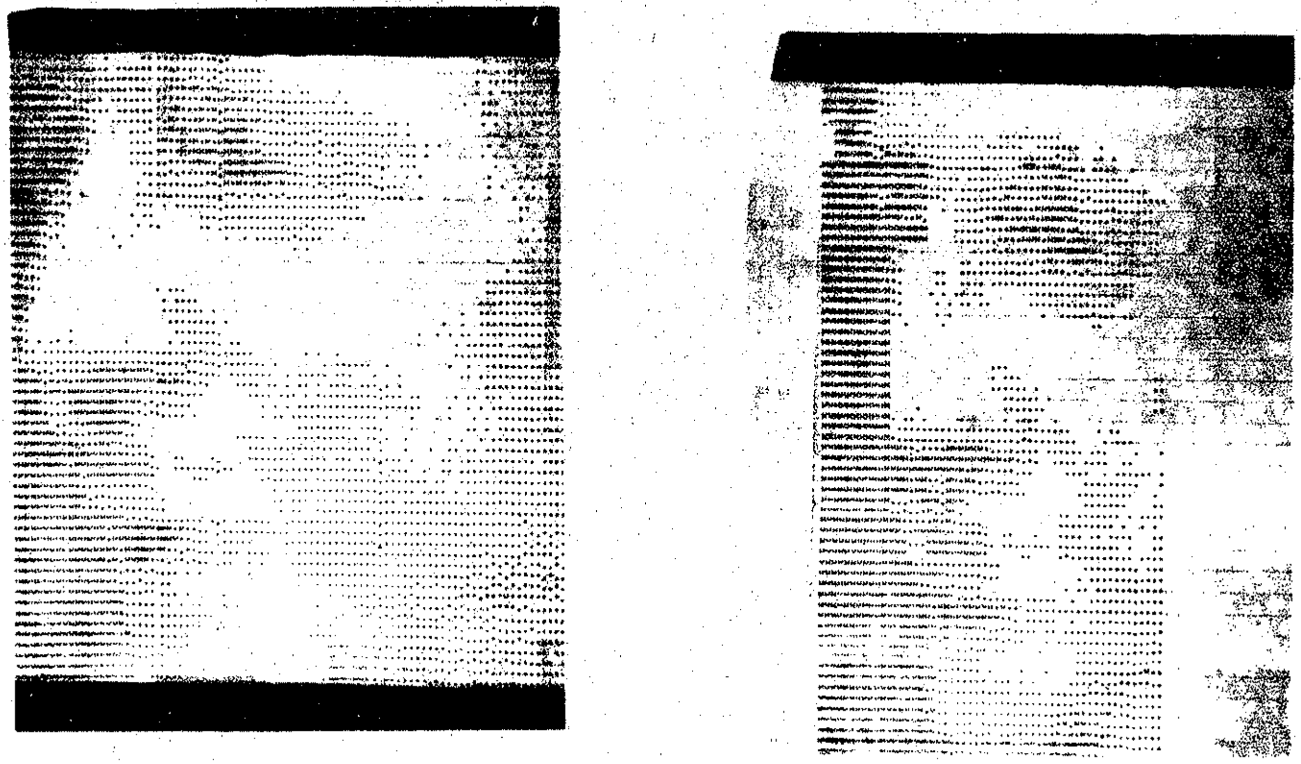


Picture 15

Figure 6A

Reconstructed 15°

Figure 6B



Reconstructed 45°

Figure 6C

Rotated 15°

Figure 6D

Figure 6

Results of Mathematical "Rotation"

The first presents no particular problem because it can be estimated fairly accurately, and trial and error will soon produce the desired result. The elevations, however, present more of a problem. One method would be to have stored and available several different typical facial structures, and then, for a new face, have the human operator select one of these as being most like the new face. Other suggestions are given in the following section.

FEEDBACK IN THE ROTATION MECHANISM.

With the proper display devices a person operating the system can be made continuously aware of the effectiveness of the rotation system. When he has put in a photograph and entered his estimates of the facial type and rotation angles, the machine will perform the mathematical rotation of the face and present the resulting picture on a cathode ray tube (or other display device). Then the operator can adjust his inputs until the resulting (rotated) picture looks right to him. At that time the rotated picture is accepted for entry into the main recognition machine. For example, the rotating machine might perform its task for a whole series of possible angles and facial types so that the human operator can pick the one producing the best results.

STEREOSCOPIC CAMERAS

The problem of rotating the faces to the frontal pose is a formidable one because of the lack of knowledge of the basic structure of the particular face that is to be recognized. Even with the feedback system described above, the operator cannot be completely sure that he has obtained the correct frontal view. This difficulty might be overcome by the use of a specially designed stereoscopic camera. Such a camera would take two pictures of the subject from two slightly different angles. The distance from camera to subject, at the time the picture is taken, can be obtained by focusing. The two pictures, along with the distance and fixed angle information, can then be used to determine the facial structure and therefrom to effect the desired rotation. Of course, the use of the telephoto lenses (for great distances) will tend to degrade the stereoscopic effect.

One way to utilize such a stereoscopic procedure would be to have the human machine operator look at the pictures through a stereoscope to estimate the facial type. It can be arranged so that he can continue to observe the rotated and reconstituted pictures through the stereoscope while the machine is rotating both pictures. This is analogous to the feedback system described above, but should be much more effective because of the three dimensional perception the operator will have during the rotation. The feedback mechanism can be arranged so that he can observe both the unrotated (original) and rotated pictures at the same time.

Of course, if we could obtain a good three dimensional representation of the face this representation itself could be used as input to the recognition machine. Thus, if for each point on the surface of the face we had the elevation there (See Figure 2) then this information could be used to identify the photo in exactly the same way that light intensity measurements from the scanner is used.

DETERMINING THE 3-DIMENSIONAL STRUCTURE OF THE FACE OF AN UNPOSED SUBJECT.

It is clear that the three dimensional structure of the face can be used as an aid in identification, either for rotation or identification. How can this be obtained for an unposed subject.

The method used above for determining the three dimensional shape of the face depended upon having two different photographs of the same subject with different angles of rotation of the head. Also, it required markings on the face (See Figure 4) so that measurements to these points could be made on the two photographs and these measurements in turn could be used to compute the three dimensional structure.

Instead of the actual marks we might use a shadow grid as indicated above. However, neither of these would be possible for an unposed subject. As one suggestion, the necessary "markings" might be obtained by using an ultraviolet light source with a stereoscopic camera. Still another method, which requires no artificial marks on the face, but utilizes instead the natural marks of the face for determining the three dimensional structure is described in the following section. Each of these methods require two photographs of the subject with different angles of rotation of the head.

DETERMINING THE THREE-DIMENTIONAL STRUCTURE WITHOUT ARTIFICIAL MARKINGS.

Suppose we have two photographs of a person taken with the same range and lighting and at nearly the same time, but with different rotation angles (See Figure 4). The following procedure is proposed for determining an elevation y at each point (x, z) of the frontal view of this face (See Figure 2).

1. Choose a reference point, O, such as the corner of the left eye which will act as the origin of the xz coordinate system.
2. Estimate the two angles of rotation θ_1, and θ_2, and the axis of rotation.
3. Arbitrarily assign to O the elevation

   $$y = y_0$$
4. Choose a nearby point $P = (x, y)$ where the elevation is

to be determined, and let R represent a small rectangle about P on the frontal view.

5. Using the techniques described above rotate the rectangle through angles θ_1, and θ_2 respectively using a fixed elevation y_0 for the rotation, and let R_1 and R_2 be the resultant rectangles on photograph 1 and 2.

6. By correlation compare R_1 and R_2. If they are the same, take y_0 as the elevation of the new point P, but if they are different -

    a. Make a slight change in y_0.
    b. Recompute R_1 and R_2, and compare again.
    c. Keep changing y_0 in a way that will maximize correlation between R_1 and R_2, until a best y(P) is found. Thus y is taken as the new elevation at point P.

7. Similarly, the elevations are determined at points near those where the elevation are already computed until all elevations are determined.

This is described more precisely as follows:

Let $I_1(x, z)$, $I_2(x, z)$ be the intensity functions for the two photographs, and suppose we have determined the elevation y at the point $P = (x, z)$. Let $P' = (x', z')$ be a nearby point, let

$x' - \delta \leq x \leq x' + \delta$, $z' - \delta \leq z \leq z' + \delta$, be the rectangle of interest.

Let

$$f(y^{o}) = \int_{x'-\delta}^{x'+\delta} \int_{z'-\delta}^{z'+\delta} \left[I_{\theta_1}(x, z) - I_{\theta_2}(x, z) \; dx \, dz, \right]$$

where $I_{\theta 1} = I_1(x \cos \theta_1 + y^{o} \sin \theta_1, z)$,

$I_{\theta 2} = I_2(x \cos \theta_2 + y^{o} \sin \theta_2, z)$.

Thus $f(y')$ represents the correlation between the two rotated rectangles if we assume a fixed elevation y' through the rectangle.

Compute $f(y)$, $f(y + \Delta y)$, find the largest of these, and adjust Δy until a maximum $f(y')$ is obtained. The elevation at the point (x', s') is taken to be the y' giving the maximum correlation. That is y' is such that

$$f(y') = \max_{\Delta y} f(y + \Delta y).$$

Thus with good estimates of θ_1, and θ_2, we can proceed from point to point until the elevations are determined for all points (x, s) on the picture.

APPENDIX V

FEATURE EXTRACTION METHOD FOR RECOGNIZING FACES

W. W. Bledsoe

Introduction.

A proposal has been repeatedly made for facial recognition by machine which would employ a system similar to that in use for fingerprint processing. Such a system would utilize various features of the face, such as nose, eyes, lips, ears, forehead, etc. Each such feature could be classified and then the whole face recognized from the combined classifications of its features. For example, the nose could be classified as long and thin, broad and short, pointed, "turned up", etc., and each of these classifications could be represented by a number (its classification number). This number would then represent the type of nose being considered. If similar classification numbers are given for the eyes, ears, and other features of the face then a complete photograph can be represented by a sequence of such numbers:

$$C = (C_1, C_2, \ldots, C_N).$$

Human Approach.

One proposal is that human observers would be used to examine a photograph and determine by inspection the classification numbers C_i, and these in turn would be "looked up" in a list to determine the identity of the person photographed.

Difficulties.

Some of the difficulties of such a proposal are as follows:

1. Human classifiers tend to be imprecise in their determination of the classification numbers, C_i, and two different observers will obtain different C_i's for the same photograph.

2. The human observer takes a great deal of time to determine the classification numbers C_i.

3. The "look up" procedure for determining the identity of the person associated with a given $C = (C_1, C_2, \ldots, C_N)$ can be formidable because of the large number of possible C's.

4. The "learning" problem for building the "look up" tables can be most difficult because of the large number of cases that have to be anticipated. This could be especially expensive and time consuming if, indeed, humans are used to determine the feature classification, C_i.

5. Lack of discrimination. The procedure tends to identify more than one person with each classification sequence, C. In fact, when only a few features are used (when N is small), one photograph can produce an unacceptable large number of possible identifications.

6. The lookup procedure can be prohibitively difficult if the number, N, of features is large.

Machine Approach.

It is thought possible that a machine could be used to take the place of the human feature recognizer and thereby eliminate most of the difficulties listed above. In particular, if a machine could be designed that would rapidly locate and classify features then the difficulty, 1, would be eliminated because the machine would repeatedly give the same response to the same picture. Also the machine speed would tend to eliminate difficulties 2-3. Furthermore, difficulty 5 could be eliminated or decreased by greatly increasing the number, N, of features without aggravating difficulty 4. Of course, if N were increased, the lookup problem would become more severe.

After such a system had been designed, setup, and was successfully operating, it would work as follows:

1. A picture of a person is received which is required to be identified.

2. The picture is transduced into a machine (by an optical scanner, for example).

3. The machine would locate the first feature (say the left eye).

4. This feature is classified by the machine by giving it a classification number C_1.

5. Steps 3 and 4 are repeated for each of the features being considered.

6. The combined features classifications

$$C = (C_1, C_2, \ldots, C_N)$$

are fed to the lookup machine which consults a list of names to determine the identity of the person in picture.

Problems to be Solved in the Machine Approach.

The workability of such a machine will depend on the successful operation of steps 2, 3, and 6.

Step 3 is the classic pattern recognition problem, as for example, the problem of recognizing printed characters by machine. Indeed, the facial recognition problem itself is similar to the printed character recognition problem, but much more difficult because of its larger variability and greater number of categories of identification. However, the facial feature classification problem is very much like the character recognition problem because of single feature, such as an ear, has much less chance for variability than a whole face, and also has fewer classification (i.e., there are fewer types of noses than there are type faces).

For these reasons it is felt that the character recognition research that has been done during the last few years and which is being done now, can be used to facilitate the solution of step 2 above. In particular, the n-tuple method for pattern recognition is a candidate for this task and has been researched extensively on large samples of variable characters (See Appendix VI).

In step 3 we encounter the problem of locating a specific feature on the photograph. This is very much like the Air Force problem of locating a target (say a missile site) on a map or an intelligence photograph. The published research has shown very little success on this type of task, but some of the new work on automatic scanning of bubble chambers may offer some hope here.[1]

Step 6 is relatively simple as long as the total number of possible facial classifications is not too large. For example, if there were only 6 features (N = 6) and each feature had at most 10 classification categories then the total number of possible facial classifications would be

$$N_F = 10^6 = 1,000,000.$$

In this case a digital computer with magnetic tapes could be used to perform a simple lookup to determine the identity. Here one would have stored in "address" C of the magnetic tape, the name of the person whose facial classification was C. $(C = (C_1, C_2, \ldots, C_N))$.

However, if the number of features was increased to 20 while the number of classification categories remained at 10 for each feature, then the total number of possible facial classifications would be 10^{20} and the simple procedure described in the above paragraph could no longer be used. In this case of many features, we are confronted with a problem surprisingly like the original one of recognising faces, because instead of recognising a complex face we must recognize the complex quantity, C. It must be realized, however, that the transformation which led from the face to the quantity, C, has very likely preserved the essential information of the face while greatly reducing the amount of information (number of bits of data) that must be processed. Therefore it is reasonable to expect that character recognition techniques could be successfully used to perform this last step of the identification. In fact, the hyperplane methods described in Appendix I or the corridor-projection presented in Appendix VII are prime candidates for this last recognition steps.

---

[1] See bibliography to the paper, "The Illinois Pattern Recognition Computer - ILLIAC III, by B. H. McCormick, IEEE Trans. on Electronic Computers, Dec. 1963, pp. 791-813.

Thus the procedure of determining the feature classification numbers, C, can be thought of as a filter which is applied to the original output from the scanner, that produces another pseudo picture, C, which now must be recognised. The masks system of Appendix I is another such filter. Pattern recognition programs (such as the hyperplane methods of Appendix I or the corridor-projection method of Appendix VII) can be used before or after one or both of these filters have been applied.

Learning.

Before this feature classification system could be put into use, it is necessary to pass through a machine "learning" phase in which the machine is shown at least one facial photograph of every person who might be required to be recognized. At the same time the machine is given the name of the person in the photograph. For each such photograph, the classification numbers

$$C = (C_1, C_2, \ldots, C_N)$$

are determined, and C is then associated with the name of the person. The name of the person is simply stored in address C on the magnetic tape (or in the computer) if the simple lookup procedure described above is being used. However, if another character recognition procedure is being used then C and the name of the person are associated in the way that is required by the particular character recognition method.

APPENDIX VI

A REPORT ON MACHINE READING OF HANDWRITTEN AND PRINTED CHARACTERS

C. L. Bisson

If facial recognition is to be accomplished by feature extraction then some of the results of character recognition are applicable. Many experiments have been run at PRI on various types of data. This data includes:

(a) Handwritten numbers
(b) Computer generated alphabets
(c) Typewritten alpha-numerics
(d) Alpha-numerics from label plate impressions

A. Results on handwritten numbers

The first attempts at recognition were performed on handwritten numerals 0, 1, ..., 9 written by fifty different people at Bell Telephone Laboratories[1] and represented by 0 and 1 on a 12 x 12 raster of points (See Figure 1) on IBM cards. The results of these tests were published.[2,3]

The source data for each character was contained in the 12 x 12 raster of points and later was centered by placing the center of gravity at a predetermined place in the raster. In the presence of background

---

[1] Provided by W. H. Highleyman and refered to by W. H. Highleyman and L. A. Kamentsky in "Comments on a Character Recognition Method of Bledsoe and Browning," IRE Trans on Elec Comp, June 1960.

[2] W. W. Bledsoe and C. L. Bisson, "Improved Memory Matrices for the n-tuple Pattern Recognition Method, IRE Trans on Elec Comp, June 1962, pp. 414-415 (Copy enclosed with report).

[3] W. W. Bledsoe and I. Browning, "Pattern Recognition and Reading by Machine," Proc Eastern Joint Computer Conference, Boston, Mass., pp. 225-232; 1959 (Copy enclosed with report).

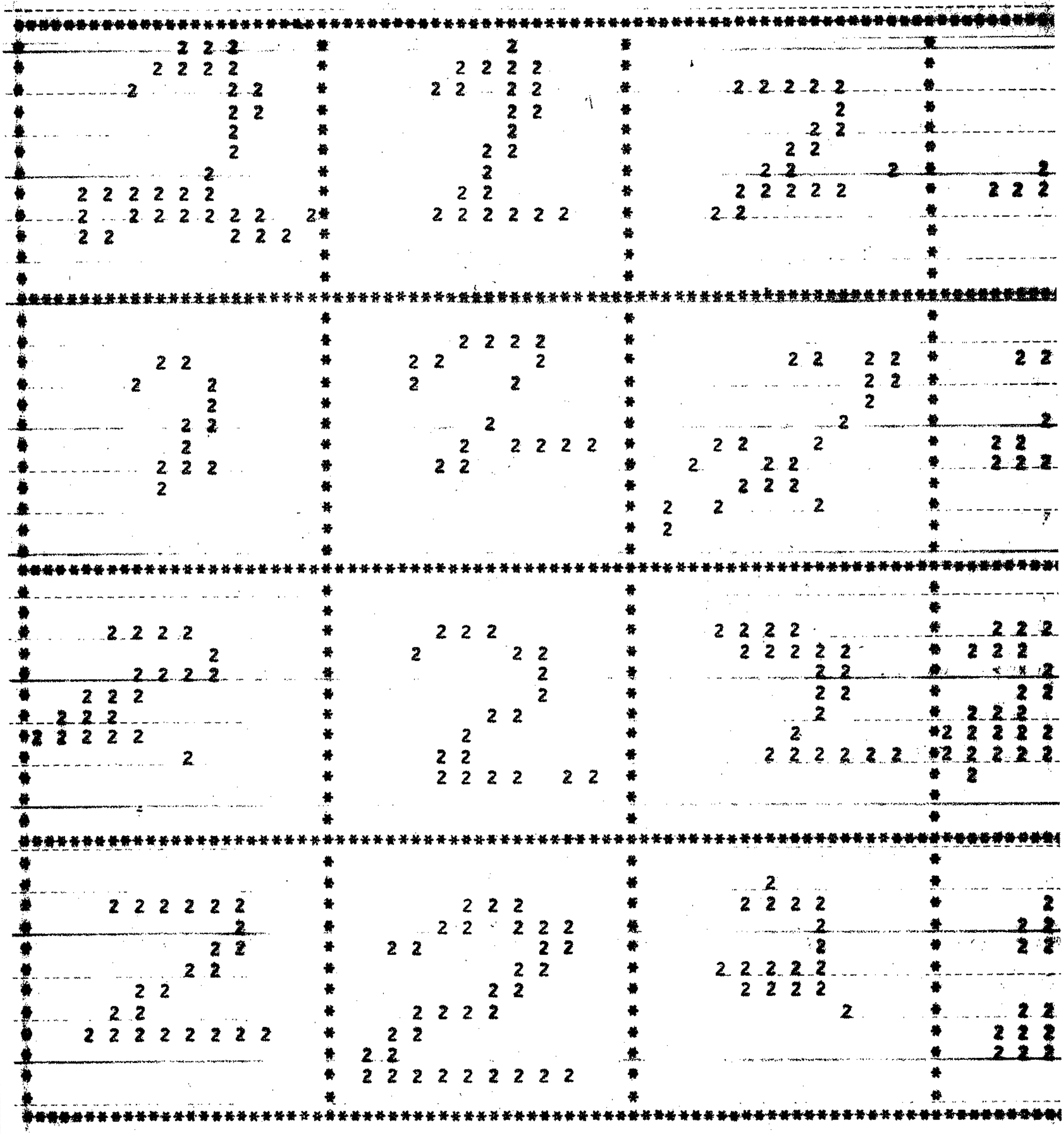


Figure 1

Examples of BTL Handwritten Numerals

noise a character can become difficult to locate or in the case of a badly mutulated character the use of the center of gravity seems questionable. Hence the problem of finding a character needed some further consideration. As was mentioned in the reference,[2] the amount of data (50 samples of each category) is insufficient for any real significant results. This lack of data would most likely not affect a comparison between two different methods but merely the quality of reading for each.

Note: The following discussion assumes familiarity with the n-tuple pattern recognition method as defined by Bledsoe and Browning.[3, 2]

The purpose of the computer runs discussed by Bledsoe and Bisson[2] was to find a memory matrix which would give the best reading ability. In this light further attempts were made to improve the memory matrix by using gradient techniques. The general approach here was to define a function, with the elements of the memory matrix as independent variables, such that when the function was maximized the reading ability of the memory matrix was also maximized for a given set of data. This was carried out and the detailed results given in the memorandum in Appendix VIII. It was found that although the readability of 100% on familiar (learned) material was obtained, readability of only 53% was achieved for new material. Thus it was concluded that larger samples of input data was needed for these experiments.

Because of the small amount of data available for research, a method of generating characters by a computer was used. The method was to specify a model character and divide it into subsets and perform statistical amounts of translation, rotation, and variation in size. The algorithms used are given in Appendix IX and sample characters generated are shown in Figure 2. A program was written to effect this algorithm and 700 A's and 700 B's were generated. These characters were later centered by using the center of gravity of the image. A series of experiments were conducted to determine the effect that increased

(400 CENTERED) 14*8 GEN FROM D1

ARD NO 1

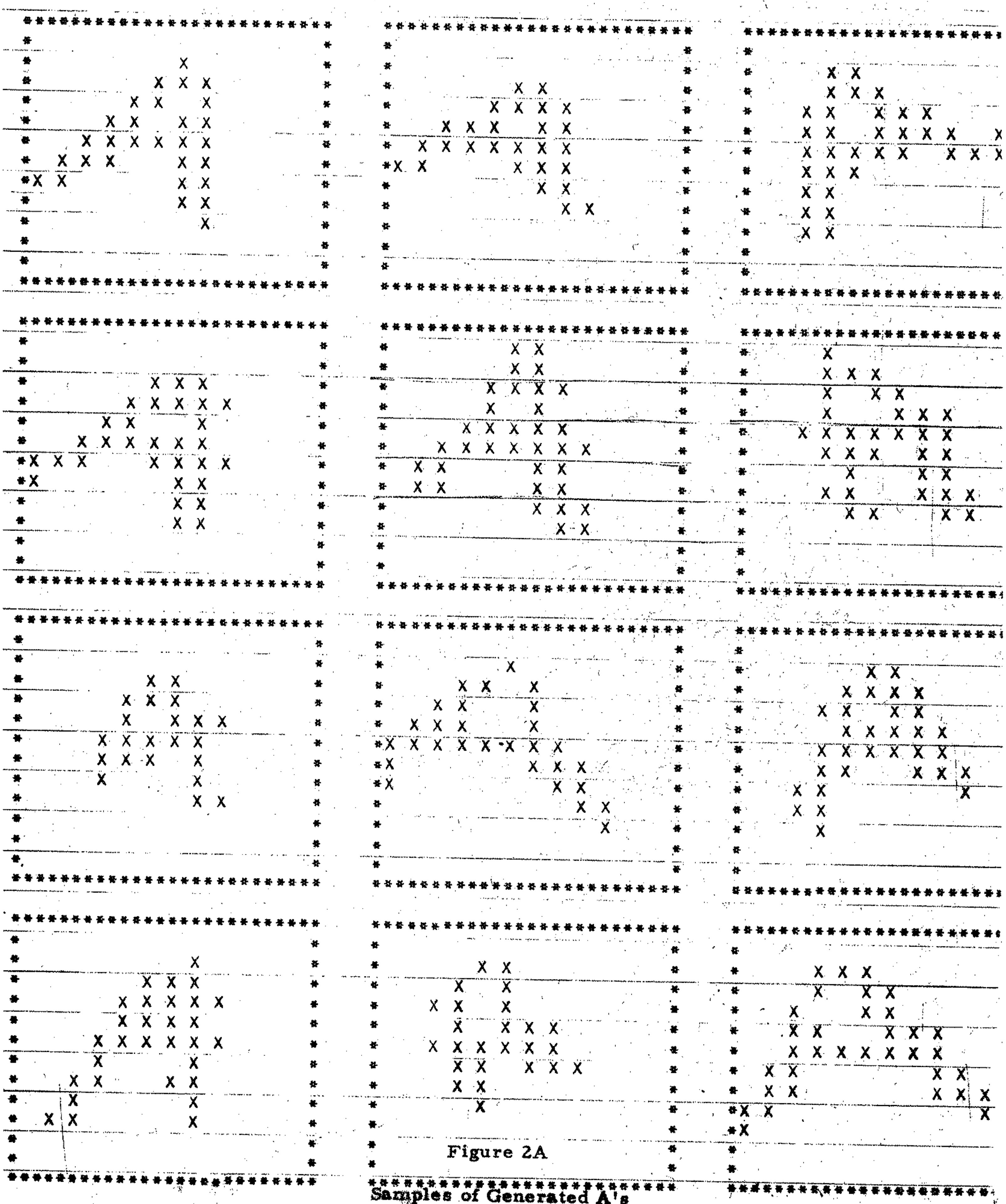


Figure 2A

Samples of Generated A's

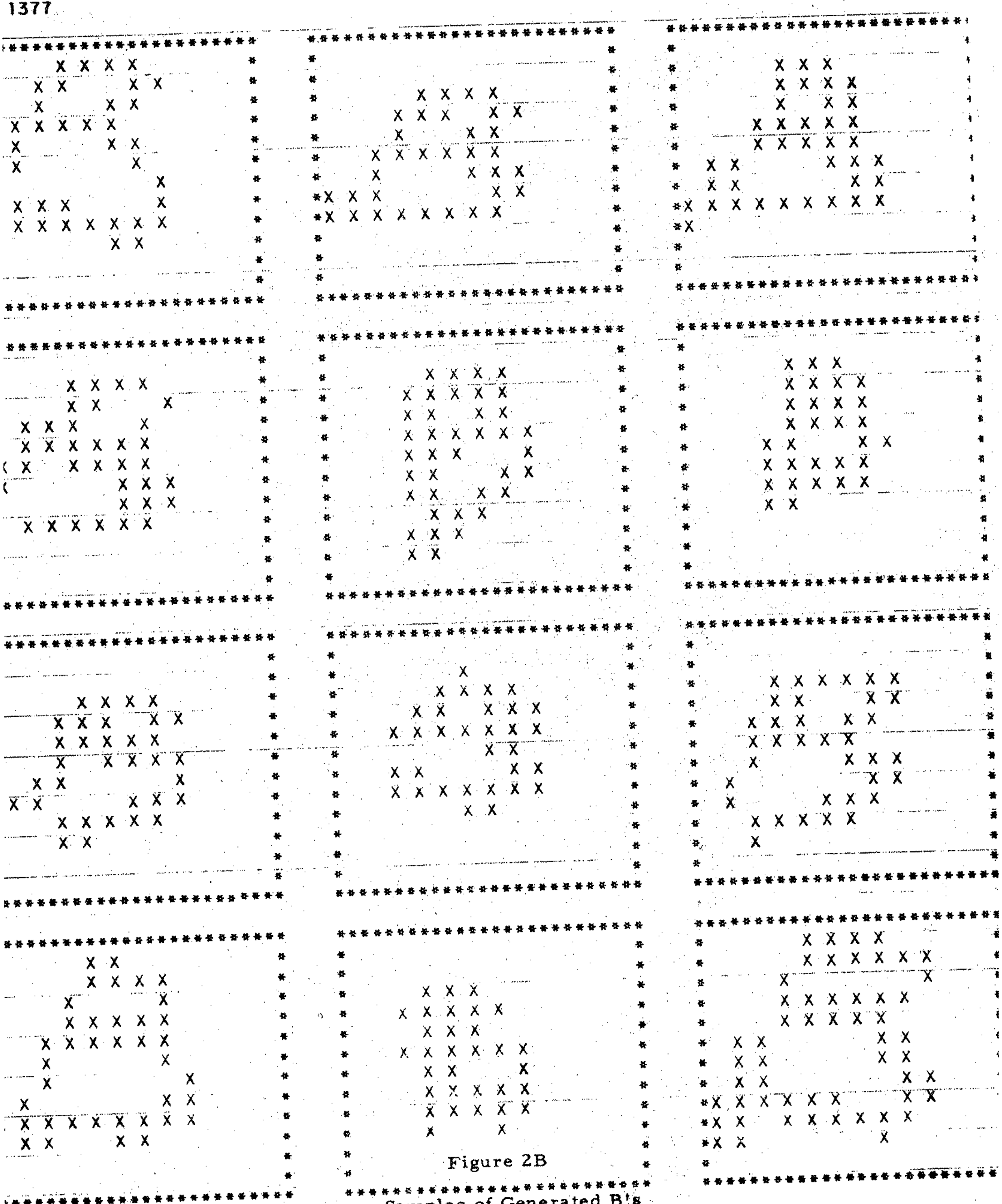


Figure 2B

Samples of Generated B's

learning would have upon reading. The readability is shown to converge to a fixed percentage as a function of the amount learned. From each set of 700 characters, 200 were set aside for test reading purposes and were never learned. From the remaining 500 characters, subsets (successively inclusive) of 50, 100, 200, 300, 400, 500 were taken and learned. Six learn and read experiments were done and in each case both the learned material and the set of 200 unlearned text characters were read. The results of these runs are given in Figure 3. The method used during the learning process was the maximum likelihood technique.[2] This method assumes that the states of n-tuples $x_1, x_2, \ldots, x_m$ where $0 \leq x_i \leq 2^n - 1$ may be considered as independent random variables. Hence if $P_{ij}(x)$ is the probability of the i^{th} n-tuple in the category j having the state x, then

$$P_j(x_1, x_2, \ldots, x_m) = P_{1j}(x_1) P_{2j}(x_2) \ldots P_{mj}(x_m)$$

and if P_q is the largest amongst the P_j, where $1 \leq j \leq Q$ (Q = the number categories). This is equivalent to choosing the largest amongst

$$\ln(P_j) = \ln \left\{ \prod_{i=1}^{m} P_{ij}(x_i) \right\}$$

$$= \sum_{i=1}^{m} \ln\left(P_{ij}(x_i) \right)$$

where ln(0) was taken arbitrarily to be $-k^2$ for this work. (Note that multiples are replaced by adds.)

It should be noted in Figure 3 that, as expected, the learned material read better than the unlearned material for both centered and non-centered characters and that the learned-unlearned curves are converging. It should also be pointed out that centering the character for learning a small

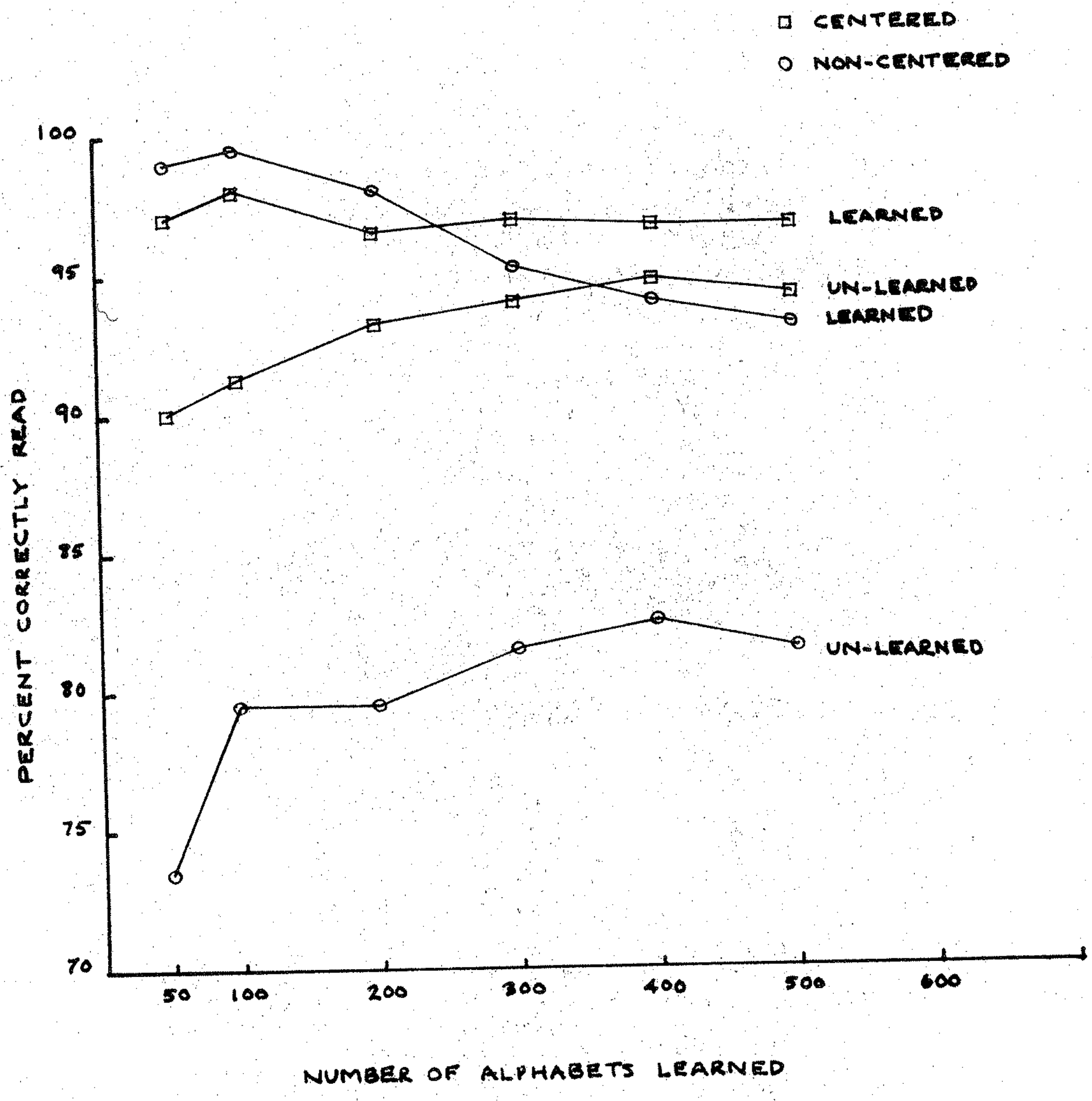


Figure 3

Percent Read as a Function of Amount Learned
(For the letters "A" and "B" only)

number of characters does not necessarily improve the reading. For small amounts of data centering makes the data less unique hence more difficult to separate. The learned non-centered and centered curves intersect after about 225 characters and for learning more than this, centering does seem to improve reading for both the learned and unlearned. This experiment clearly shows the advantages of some centering technique and increased samples for learning. Any further experiments involving this data was not attempted because the destortions of some of the characters were too extreme. See Figure 2 for samples of the generated characters.

Two new sources of data, kindly furnished by the Sandia Corporation, New Mexico, were obtained by optical scanning. This data was not, however, handwriting but typewritten data. The first set of data was typewritten alpha-numeric represented in a 20 x 30 raster with one character per IBM card and approximately 20 samples of each character. It was felt that the 20 x 30 raster was too fine for the character, i.e. highly redundant and they were of good quality, so only one-quarter of the bits were used by choosing bits from alternate rows and columns thus giving essentially a 10 x 15 raster. Again the recognition scheme was the n-tuple method using several values of the n-tuple size and m, the number of n-tuples used on the raster. Runs were made on numerals only, and numerals and alphabets, and the results are given in Table 1.

Notice that although the results of these runs were encouraging they were not realistic. That is to say, the small sample size did not represent data that was obtained from poor impression, mutilated characters, variation in intensity etc. as was to be the case when large samples of characters were obtained.

B. Label reading study

In this study a new source of machine printed data was introduced that was plentiful, available, represented a wide variety of quality and

Table 1

Typed Alphabet Results

n = 5 m = 30

Remarks	No. of Categories	No. Characters Used		Percent correctly Recognized	
		Learning	Reading	Learned	Unlearned
Numbers only, (0, 1) Memory Matrix	10	10	20	100	96
		20	20	100	--
Alphanumeric (0, 1) Memory Matrix	36	20	20	100	--
Numbers only, Max. Likelihood Memory Matrix	10	10	20	100	91
	10	20	40	97.5	93.5

was a challenge in that font style was clearly not intended to be read by machine. Samples of this data are shown in Figure 4 and are impressions made from a form of Addressograph plates. A total of 41 different type of characters are possible which include the alphabets, numbers and some special characters. It should be noted that letter Ø and the number zero are identical as also are the letter I and the number one. Thus, if this identity problem is to be solved, context becomes necessary and in most cases this could easily be accomplished. The printing is in the format of names and addressed for mailing purposes and can have as many as five lines of 24 characters each but averages to about 3 lines with 16 characters. These impressions were made on white cards with one label per card.

The first problem was to get the sourced data scanned and in a form that a digital computer can handle. In view of the large amounts of data to be processed digital magnetic tape was best suited for this purpose. Hence through the efforts of W. E. Evans and H. E. Murphy of the A. B. Dick Company (now with Granger Associates) a flying spot scanner was provided to write the video scanned and digitized on magnetic tape using an IBM Model 729 IV tape transport. Each line of scan was written as 144 characters (6 binary bits/character) on tape as one logical record.

The first attempts at scanning and reading were made such that a character represented by an 8 x 8 raster. It was soon discovered that this scan density was too course. That is, in some cases there were simply not enough bits by which a decision could be made even though in the majority of cases it was sufficient. As a result the scan density was almost quadrupled yielding a scanned character in a 15 x 15 raster and the computer could extract the 8 x 8 raster as before and proceed as before. When it was not possible to recognize a character using the 8 x 8 extracted raster it was possible to go back for more bits which would hopefully make a decision possible.

MRS FOREST HOSLER
RIVES JUNCTION
MICH

11 C563 PP 6K1 10-3

MRS MAGDALENO MENDEZ
8493 RIVES JUNCTION RD
RIVES JUNCTION MICH

[illegible] J21 [illegible]

Figure 4

Samples of Addressograph Labels

Again the recognition scheme used was the n-tuple method with $n = 8$ (the size of the n-tuples), $m = 8$ (the number of n-tuples) which were taken both in the horizontal and vertical directions using only every other source bit and the elements in the memory matrices were binary.

In this system of learning the memory matrix is not concerned with statistical frequencies, etc. but merely whether or not a particular character has ever been learned or not. Hence every effort must be made to keep characters that were extremely poor representatives of their category or miss-named characters from being learned. This required that each of the approximately 15,000 characters which were used in the learning phase, had to be checked by human observers.

The n-tuple recognition method satisfactorily recognises about 80-90% of the characters read and the remaining characters are read by additional routines that are called Tiebreakers. When the n-tuple schemes cannot recognise a character it provides a list of tied categories $i, j, \ldots, k$. This list could contain from 2 to 5 members, and in most only 2. Thus the number of possible categories has been reduced from 40 to 2. The pairs that do tie-up are quite consistent and happily this reduces the number of possible ties or pairs that must be considered from the maximum possible of 40^2 to about 60 or 70. The methods used to break ties do not depend upon learning hundreds of parameters but only 2 criteria $L(i, j)$ and $U(i, j)$ for each pair (i, j), that can quickly be determined. Each tie-breaker routine makes measurements on the character raster and computes a scalar valued function $T(i, j)$. Some of the measurements made are the amount of black in a certain area, moments, the extent of a boundary etc. From the list of tied categories $i, j, \ldots, k$ a pair is chosen say (i, j). If we denote the answer for the pair by C then one of the following occur:

1. If $T(i,j) < L(i,j)$ then $C = i$
2. If $L(i,j) \leq T(i,j) \leq U(i,j)$ no decision can be made.
3. If $T(i,j) > U(i,j)$ then $C = j$.

If more than two categories are to be broken the process is continued until either one category is found that pair wise beats every other category in the list in which case C is the final answer or the list is exhausted and we cannot find the answer and the character being considered is rejected.

The method of finding a character represented in a raster of width w and depth d located at a predicted position (x_p, y_p) were solved by finding the boundaries of the character in the vertical and horizontal directions. The boundaries of the character is defined by three consecutive black (at least one bit is black) preceded by at least one white (no bits). The character is first located in the vertical direction by finding the top boundary, bottom boundary and then in the horizontal direction finding the left boundary and then the right. Using the distances found relative to (x_p, y_p) the character is centered in the horizontal and vertical direction by choosing the proper (x, y) of the raster. If the character cannot be located for some reason a bit count is performed and if there are insufficient bits for a character it is treated as a blank and advances to the next position. If the resulting bit count is not small enough to be considered noise, then the character is attempted to be read as is in the predicted position (x_p, y_p). Once a character has been found and recognized the position of the next character is predicted from the one just read and thus the reading is somewhat independent of character alignment.

The results of the label recognition study are given in Table 2. Samples of the source data are given in Figure 4 and samples of the transduced data are given in Figures 5-7. Some of the characters in

```
                                 000000000
                               00000000000            000
      000        000           00000                 0000
      0000       0000          0000                  00000
      0000      0000           0000                  00000
      0000      0000           0000                  00000
      00000     0000           0000000               0000
      00000    00000           0000000               0000
      0000000  00 00           0000                  0000
     0000 000000  00           0000                  0000
     0000  00000  00           0000                  00000
     0000  00000  000          00000                 000000
      000   000    00          000000000000          000000000000
                               000000000000           000000000000
                                  0000                 00000

NPLATE=212 NPLATI=   3   0   0   1

   2   3   4   0   0   1 212 101   0


                                                          0000000
                                 0000   0000            0000000000
           00         000         000    0000          00000    00000
          0000       0000        000       0000        000           000
          00000      0000        0000      0000        0000
          00000      0000        00000    0000          00000
          00000     0000         00000000000             000000000
          000000    0000         00000  0000                00 000000
          000000  00000          0000     000                    0000
          00000000000            0000     000           0          000
          0000 00000             0000      000         0000      0000
          000  00000             000       000         00000000000000
          000  0000              000       0000         0000000000
                                                          0000000


                                                              0000000
                 000                     000                0000000000

NPLATE=213 NPLATI=   4   0   0   1

   2   3   4   0   0   1 213 101   0

                                   000                      00000
          00        000           00000000               0000000000
         0000       0000         00000  0000             00000000000
         00000      0000         000       0000         0000      0000
         00000     00000         00        0000        00000       000
         000000    00000         000       000         00000
         0000000   00000         00000000000            000000
         0000000  000000         0000000000               0000000000
         00000000 00  00         000   0000                 0000000000
         0000 000000  00        0000    0000                     00000
         0000 00000   00        0000     000            00         000
         0000  0000   000       0000      000          00000       000
         000    00    00         00       000           0000000  00000
                                           00           00000000000
                                                          00000000
```

Figure 5

Samples of Scanned Data (15 by 15)

```
                                                             0000
NPLATE=214 NPLATI=   5   0   0   1
```

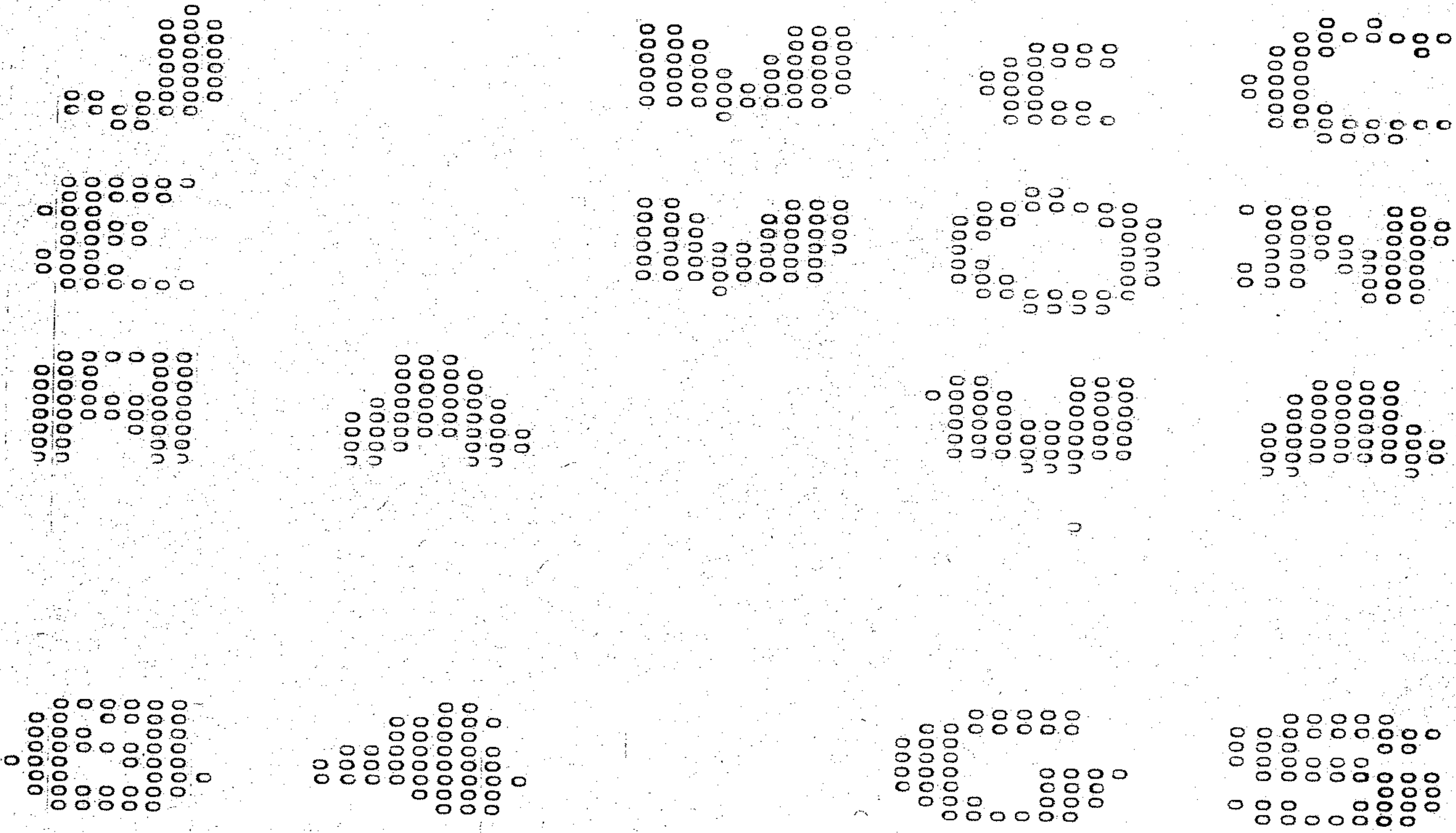


Figure 6

Samples of Scanned Data (8 by 8)

```
BOX EDIT  900   1  30  30  19
NBOX=   1  0  0  0
00000000000000000000000000000000000000000000000000000000000000000000000000
0                    0  0                     0  0                    0  0
0                    0  0                     0  0                    0  0
0                    0  0                     0  0     0000000        0  0
0     0              0  0                     0  0   0000000000       0  0
0   0000             0  0       0             0  0   00000            0  0
0  00000             0  0    0000             0  0   0000             0  0
0  0000              0  0   00000             0  0   0000             0  0
0  0000              0  0   0000              0  0   00000000         0  0
0  00000             0  0   0000              0  0   00000000         0  0
0  00000             0  0   00000             0  0   00000            0  0
0   0000        00   0  0   00000             0  0   0000             0  0
0   00000     0000   0  0    0000         00  0  0   0000             0  0
0    00000   00000   0  0    00000      0000  0  0   00000            0  0
0     00000000000    0  0     00000    00000  0  0   00000000000000   0  0
0      00000000      0  0      00000000000    0  0  0000000000000000  0  0
0         000        0  0       00000000      0  0   00000000000000   0  0
0                    0  0          000        0  0   00000000         0  0
0                    0  0                     0  0                    0  0
0                    0  0                     0  0                    0  0
0000000,,,,,,,,0000000000000000,,,,,,,,0000000000000000,,,,,,,,0000000000
  1   1   16              1   1   16               1   1   20
NBOX=   6  0  0  0
00000000000000000000000000000000000000000000000000000000000000000000000000
0                    0  0                     0  0                    0  0
0                    0  0                     0  0                    0  0
0                    0  0                     0  0                    0  0
0    00       000    0  0                     0  0    0000    000     0  0
0   000        0     0  0    0          00    0  0   0000      000    0  0
0   0000             0  0    00        000    0  0   0000             0  0
0   00000000         0  0    0000     0000    0  0   0000             0  0
0     0000000000     0  0   000000000000      0  0    0000  0000      0  0
0            00000   0  0    0000    000      0  0           00000    0  0
0             0000   0  0    000      000     0  0            00000   0  0
0   00         000   0  0    00       000     0  0             0000   0  0
0  00000      0000   0  0   000       0000    0  0   000        000   0  0
0   0000000 00000    0  0   000       00000   0  0   0000      0000   0  0
0     0000000000     0  0    00        0000   0  0    00000000000     0  0
0      000000        0  0                0    0  0       0000000      0  0
0                    0  0                     0  0                    0  0
0                    0  0                     0  0                    0  0
0                    0  0                     0  0                    0  0
0                    0  0                     0  0                    0  0
0000000,,,,,,,,0000000000000000,,,,,,,,0000000000000000,,,,,,,,0000000000
  2   1   17              8   1   2                8   1   12
NBOX=  11  0  0  0
00000000000000000000000000000000000000000000000000000000000000000000000000
0                    0  0                     0  0                    0  0
0                    0  0                     0  0                    0  0
0                    0  0                     0  0                    0  0
0            000     0  0   0000000     000   0  0                    0  0
0            0000    0  0  00000000000000     0  0             0      0  0
0            00000   0  0  00000000000000     0  0   000     000      0  0
0           00000    0  0  00000  0000000     0  0   00000  0000      0  0
0         000000     0  0 00000    000000     0  0   00000   0000     0  0
0       0000000      0  0  0000    00000      0  0   000      000     0  0
```

Figure 7

Samples of Some Poor Characters

Figure 7 were rejected while reading. For the results given, the total learning experience was approximately 600 plates (labels). In Table 2 the results are given for reading 600 plates, the first 300 of which were learned. It should be noted that quite frequently the printing of the top line was cut off at the top of the characters (See Figure 7).

	All Labels	Without Badly Mutulated Labels
Rejects	10%	5%
Errors	7%	2.2%

Table 2
Results of Reading Labels

Areas for Study

Context was used somewhat in the original presentation of the n-tuple method[3] and is an extremely valuable aid in both character and facial recognition. There have been characters in the label recognition experiments that could not be recognised by the methods discussed and when a simple rule of context was added many of these were correctly recognised. Consider, the following example: Suppose in reading the word BOX we could not read the alphabet B because it looked too much like the number 8. Hence we either have BOX or 8OX. The simple rule applied here is obvious. If the remaining characters in a word are all alphabets then the alphabet (B here) is chosen. If, however, all the remaining characters in the word are numbers, the number is chosen. Now certainly there are instances in which this rule will fail but for the majority of cases it will be correct. In the case of handwriting perhaps the development of context algorithms will be the only possible solution and certainly will not be as simple as the one above.

The results of the label recognition study seem to indicate that character recognition of a fixed font at high speed is feasible using a general purpose digital computer.

APPENDIX VII

A CORRIDOR-PROJECTION METHOD FOR DETERMINING ORTHOGONAL HYPERPLANES FOR PATTERN RECOGNITION

W. W. Bledsoe

9 October 1963

1. Introduction. Hyperplanes are widely used in pattern recognition research[1] as a means of partitioning the pattern space to achieve recognition. For this a pattern is transduced into a sequence of measurements, $x = (x(1), x(2), \ldots, x(n))$, which can be visualized as a point in n-dimensional Euclidean space. This transduction is usually done by optical scanning followed (or proceeded) by various filters (non-linear or not).

Since the patterns are in this way representable as n-dimensional points, the pattern recognition problem becomes one of partitioning n-dimensional space in such a way that each cell of the partitioning contains at most one category of points (patterns).

Because of their inherent simplicity hyperplanes are employed extensively as partitions. (See Figure 1) The hyperplanes which are used are usually determined for a collection of known images and then tested on unknown images to determine their effectiveness.

---

[1] See, for example,
Highleyman, W. H., "Linear Decision Functions, With Application to to Pattern Recognition," PROC IRE 50, 6 (June 1962), 1501-1515.

B. Widrow, "An Adaptive 'Adaline' Neuron Using Chemical 'Memistors'," TR No. 155-3-2, Stanford Electronics Lab., 17 October 1960.

O. Firschein and M. Fischler, "Automatic Subclass Determination for Pattern-Recognition applications," IEEE Trans. Elec. Comp., April 1963, pp. 137-141.

C. K. Chow, "An Optimum Character Recognition System Using Decision Functions," IRE Trans. on Elec. Comp., Vol. EC-6, pp. 247-254, Dec., 1957.

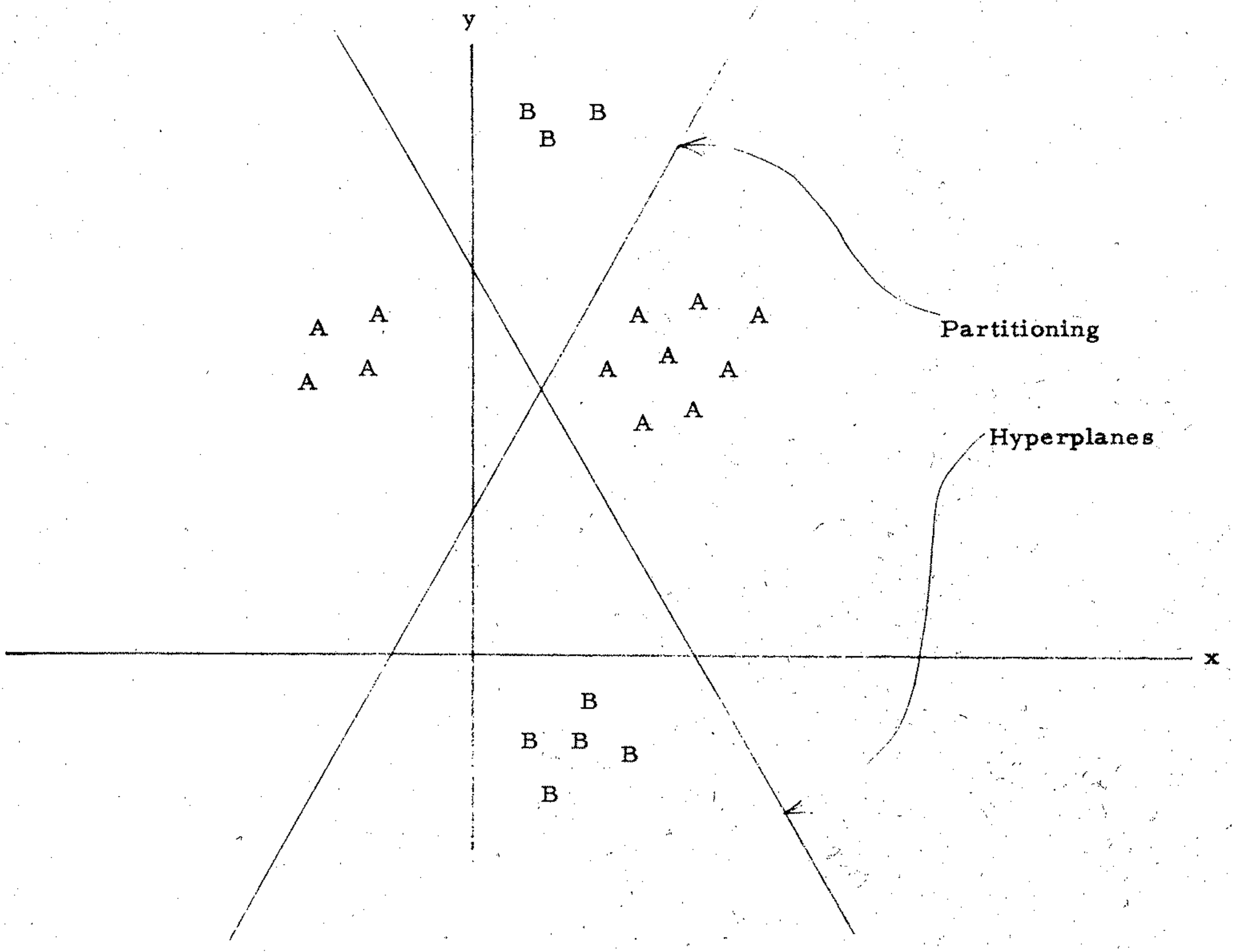


Figure 1

Partitioning of the Pattern Space
by two hyperplanes

One procedure is to determine a partitioning hyperplane for each pair of categories (i.e. A vs B, A vs C, B vs C, etc.). Thus, if there are Q categories, a total of $\frac{Q(Q-1)}{2}$ hyperplanes would be determined. The shortcomings of this procedure are as follows:

(1) One hyperplane may not be an effective partition between two categories when the images of one category are scattered and/or bimodial.

(2) Whenever the number, Q, of categories is large, the number of hyperplanes $N = Q(Q-1)/2$ is prohibitively large.

Since each hyperplane makes a binary decision for each pattern, a total of N hyperplanes will produce an N bit binary number $B = (b_1, b_2, \ldots, b_N)$, thereby converting an n-dimensional point (pattern) into an N-dimensional one. Even if this partitioning is completely effective, in that every category is partitioned from every other one, the problem is still not solved if N is large, because these new N-dimensional patterns will themselves have to be recognized by a new pattern recognition device.

A desirable objective, then, is to select a minimum number of hyperplanes which effectively partition the space. This seems to require a kind of "orthogonality" or "independence" between the hyperplanes. With this in mind we describe below a "corridor-projection" method for selecting hyperplanes. We shall designate a hyperplane by its normal vector, A. (See Figure 2)

II. General Procedure. The procedure is given in seven steps. These are first given in general terms and repeated in some detail later.

Step 1. A collection of m known points (patterns), called the "learning set", is given. (See Figure 3)

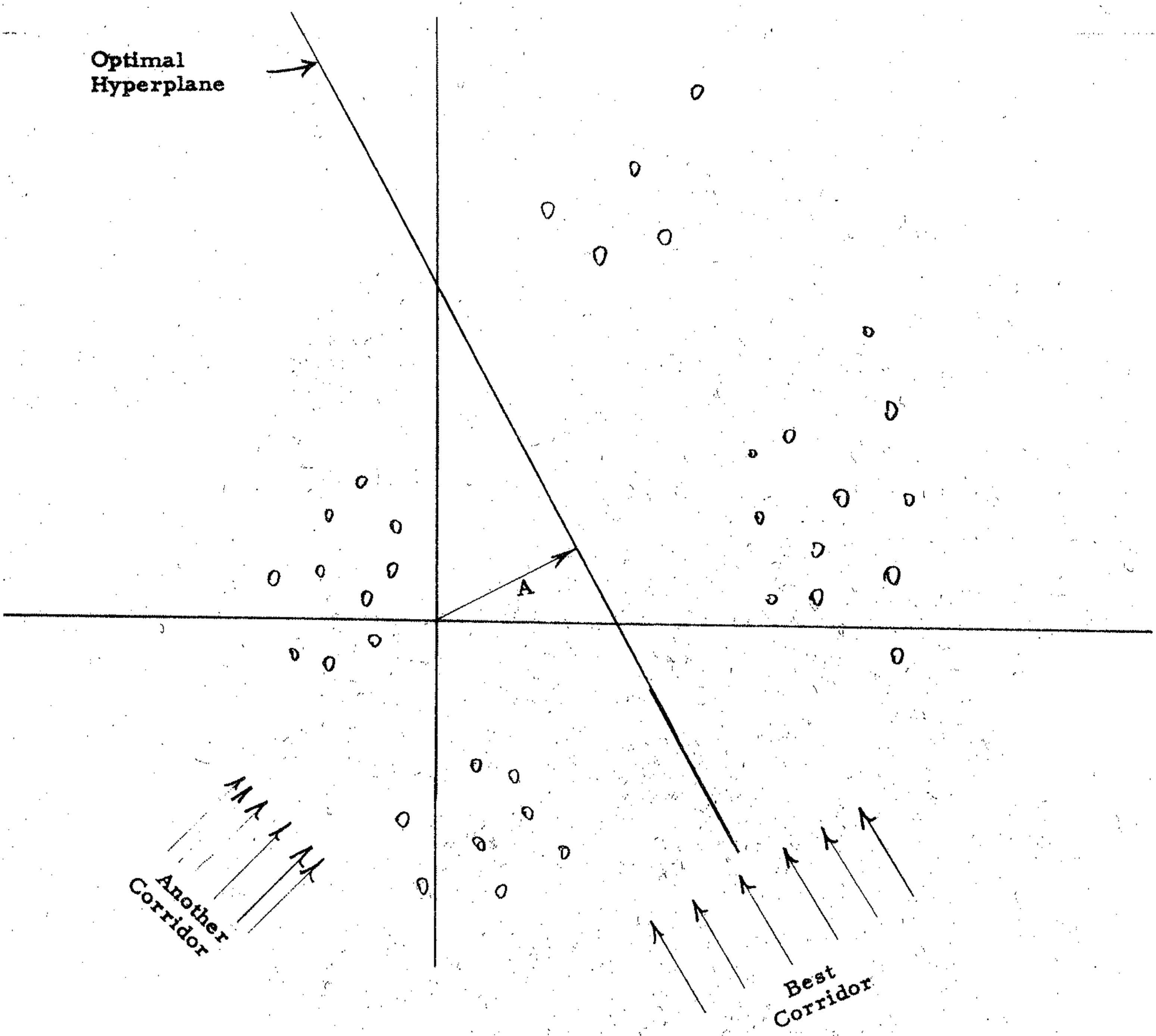


Figure 2.

Best corridor in the learning set, and the optimal hyperplane A.

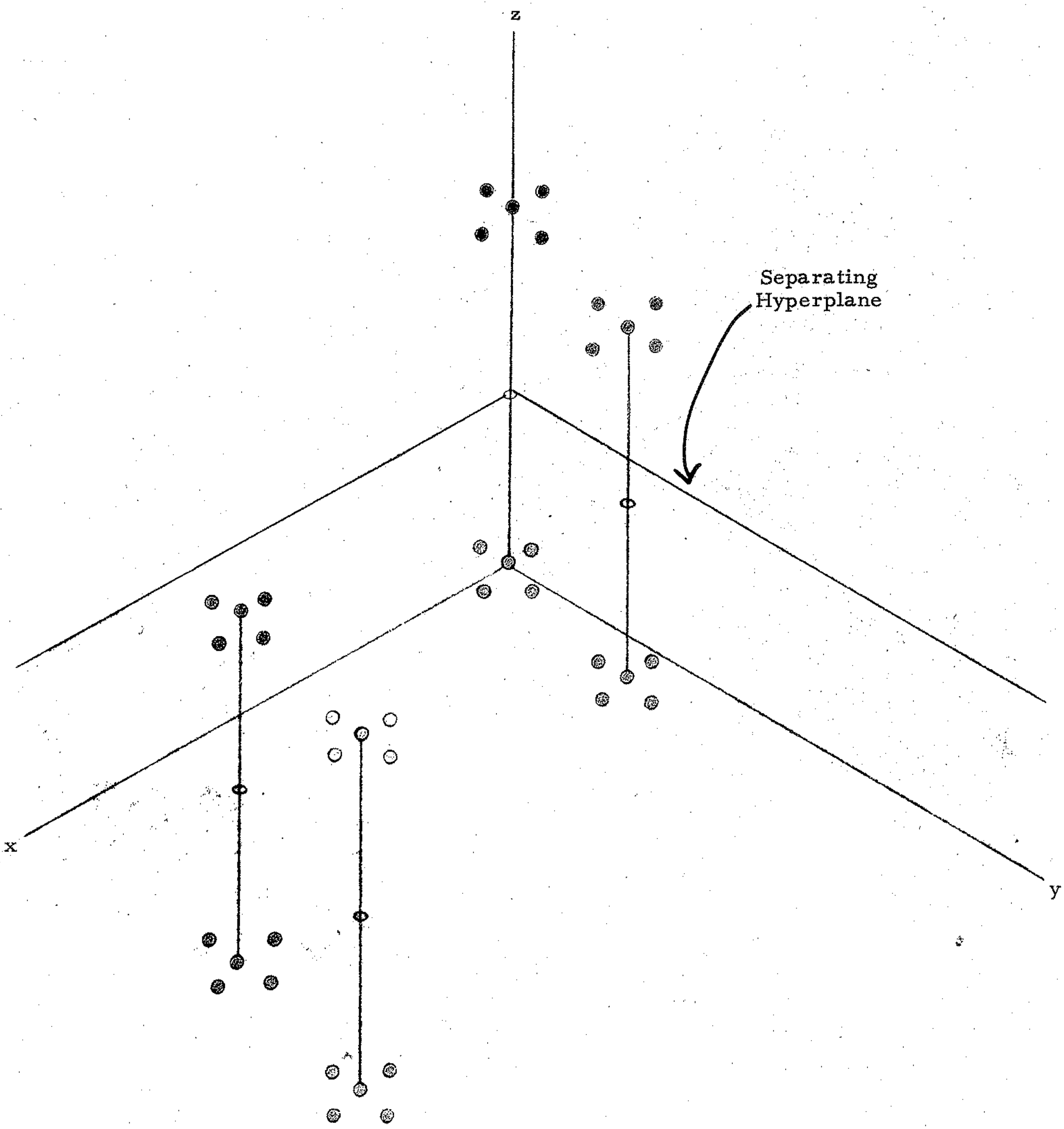


Figure 3

Three Dimensional Space with Learning Set and Separating Hyperplane

Step 2. A "best" corridor through the learning set is found along with an optimal hyperplane passing through this corridor. (See Figure 3)

Step 3. The space is rotated so that the normal, A, to the hyperplane becomes the n^{th} axis.

Step 4. The (new) points are projected upon the hyperplane by dropping the n-th coordinate. Thus the learning set has been transformed into a set in n-1 dimensional space. (See result in Figure 4)

Step 5. Steps 2, 3, and 4 are repeated for a desired number of times or until a specified result is obtained. (See Figure 5)

Step 6. The hyperplanes A_1, A_2, ... found by repeated application of steps 2, 3, and 4 are rotated back to their natural positions in the original (n-dimensional) pattern space.

Step 7. The effectiveness of these hyperplanes is tested by applying them to the original learning set.

III. Selection of Corridors. The following general method for evolutionary hill climbing, which was applied to this specific problem, is given in steps (1) - (11) below. Recall that the learning set consists of the m points: $x_1, x_2, \ldots, x_m$.

(1) Given a base A_{oo}:

(2) Select a trial-base, A_o, by putting

$$A_o(j) = A_{oo}(j) + \sigma_B \cdot RN; \quad j = 1, n,$$

where $\sigma_B \cdot RN$ is a random number chosen from a normal distribution with variance σ_B^2.

(3) Select a point A' by putting

$$A'(j) = A_o(j) + \cdot RN; \quad j = 1, n.$$

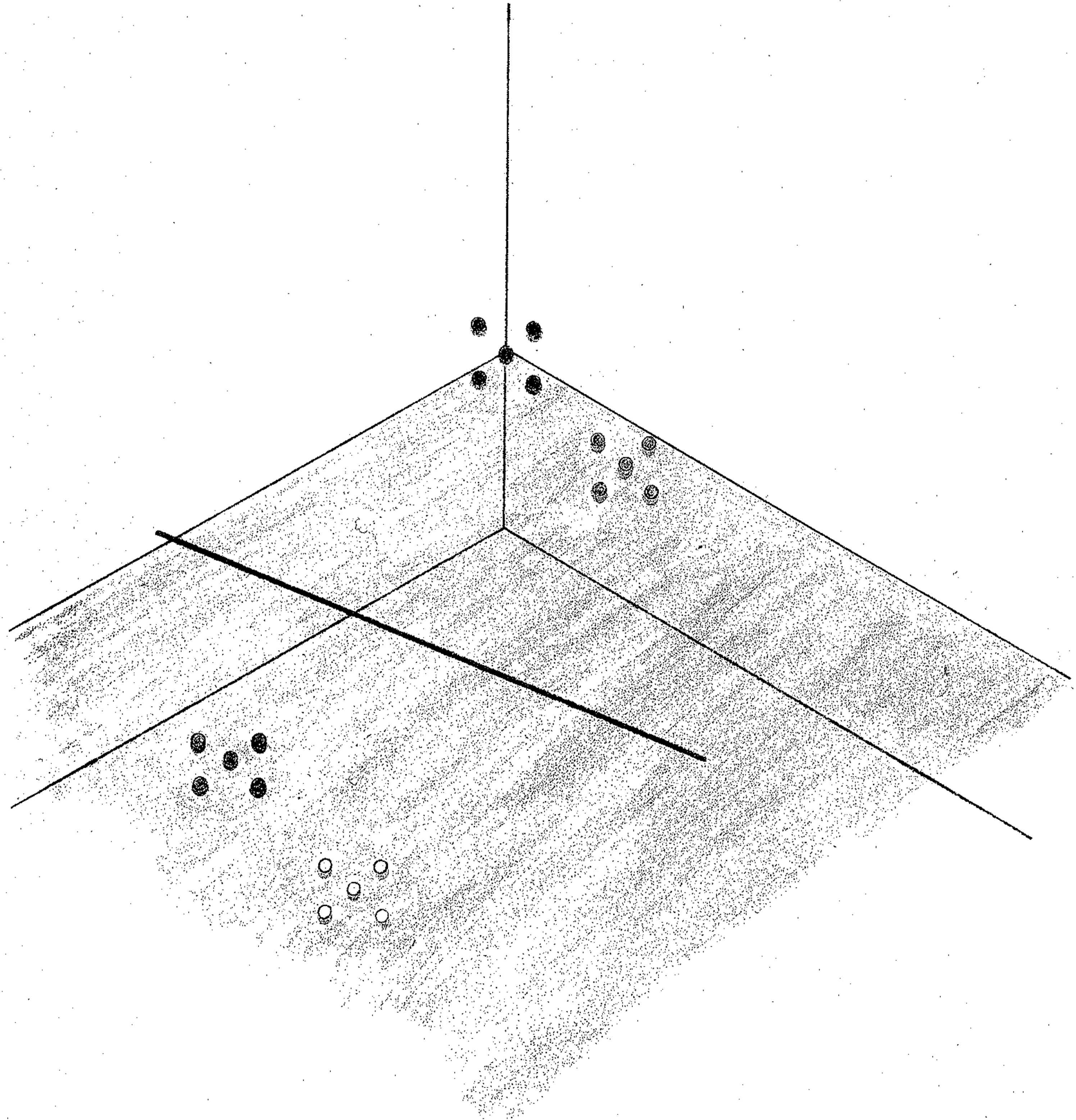


Figure 4

Learning Set After Their Projection on
Two-Dimensional Hyperplane.

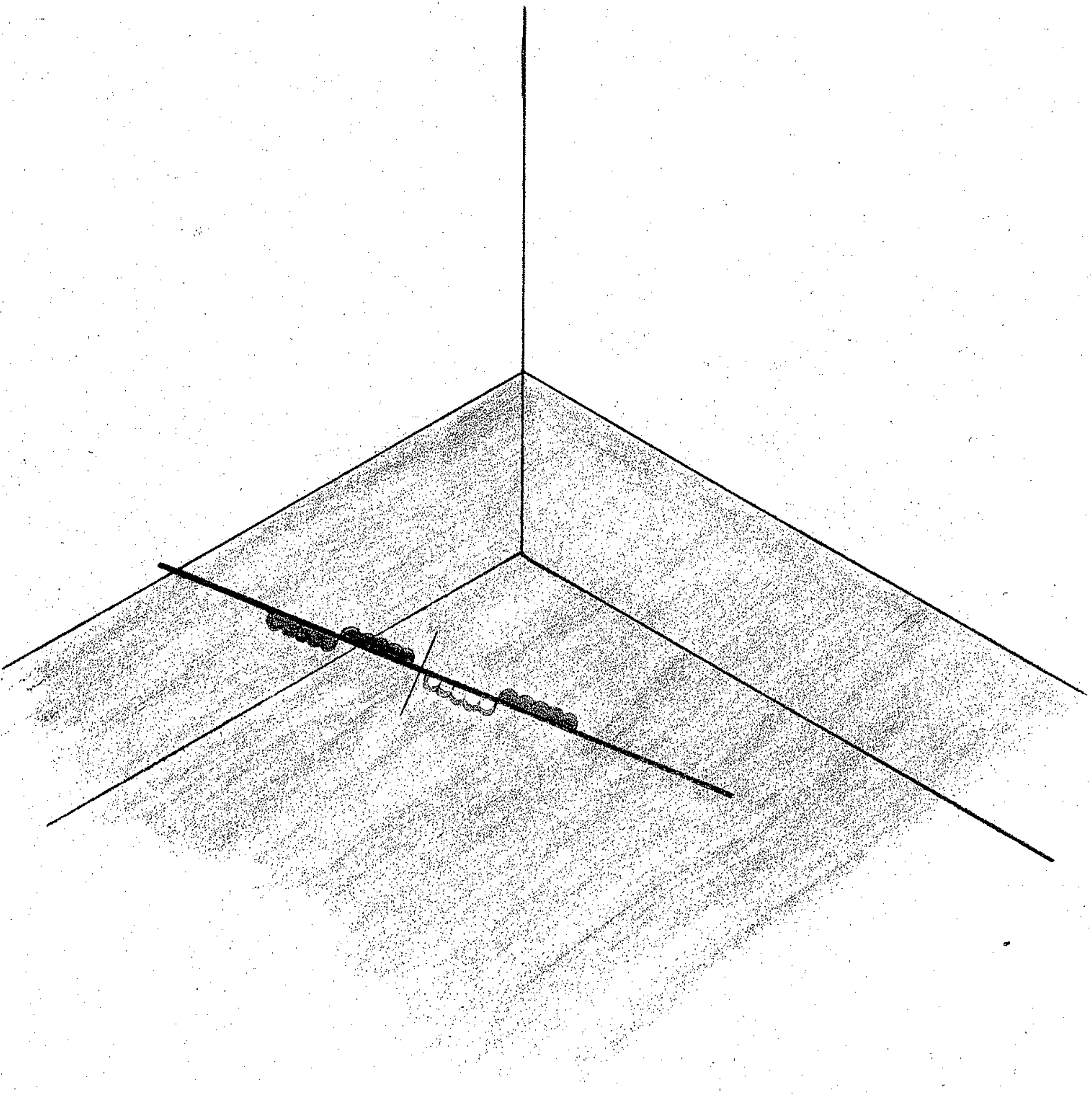


Figure 5

Learning Set Projected Further to One-Dimensional Line

(4) Adjust the hyperplane A' to its optimal position, A, (in whatever corridor it happens to lie) by using a gradient technique to minimize the function

$$T(A) = \sum_{i=1}^{m} 1/\left[D(x_i,A)\right]^2 .$$

where $D(x_i, A)$ is the distance from the hyperplane, A, to the point x_i. That is

$$D(x_i, A) = \left(\sum_{j=1}^{m} x_i(j) \; A(j) - 1\right) / \left|A\right| .$$

where

$$\left|A\right|^2 = \sum_{j=1}^{n} \left[A(j)\right]^2 .$$

(5) Find the width, W(A), of this corridor, by

$$W(A) = 2 \cdot \underset{i=1,\, m}{\text{minimum}} \left|D(x_i, A)\right| .$$

(For hyperplanes, A', which are not centered in their corridor, W(A') is defined to be the width, W(A), of a centered hyperplane in that corridor.)

(6) IF $W(A) \leq W(A_o)$, proceed to step (7), but if $W(A) > W(A_o)$ proceed to step (8).

(7) Repeat steps (3) - (6). The number of times that steps (3) - (6) are repeated for a fixed A_o is limited to a given maximum. IF this maximum is exceeded go to (9).

(8) Put $A_o = A$ (i.e., move the trial-base to the better position) and repeat steps (3) - (7). After repeating step (8) a given maximum number of times, proceed to step (9).

(9) IF $W(A_o \leq W(A_{oo})$, proceed to step (10), but if $W(A_o) > W(A_{oo})$, put $A_{oo} = A$. and proceed to step (10).

(10) Repeat steps (2) - (9). After repeating steps (2) - (9) a given maximum number of times, for a fixed A_{oo}, proceed to step (11).

(11) The corridor in which A_{oo} lies is taken to be the "best" corridor, and A_{oo} is the projection plane for steps 3 and 4 of section 2.

It should be noted that the trial-base, A_o, selected in step (2) above is kept and used in subsequent steps even though $W(A_o) \leq W(A_{oo})$, but, on the other hand, the trial-base A_o is not changed to a new position A unless $W(A) > W(A_o)$. In this way a trial starting base can be set up in step (2) which might lead to a final trial-base, A_o, for which $W(A_o) > W(A_{oo})$, and in this case the base A_{oo} will be moved to this better A. (See Figure 6)

A further refinement of this procedure is to gradually increase the variances σ_B^2 and σ^2 when, and if, the newly selected trial-bases do not produce larger corridor widths. Also the corridor W(A) could be defined in a variety of ways to reflect the density or sparsity of point about the hyperplane.

To prevent the procedure from finding the large infinitely wide "corridor" on the outside of the learning set, in step (5) the hyperplane, A, is rejected and the procedure is directed back to step (3) if hyperplane, A, divides the learning set into two grossly unequal parts.

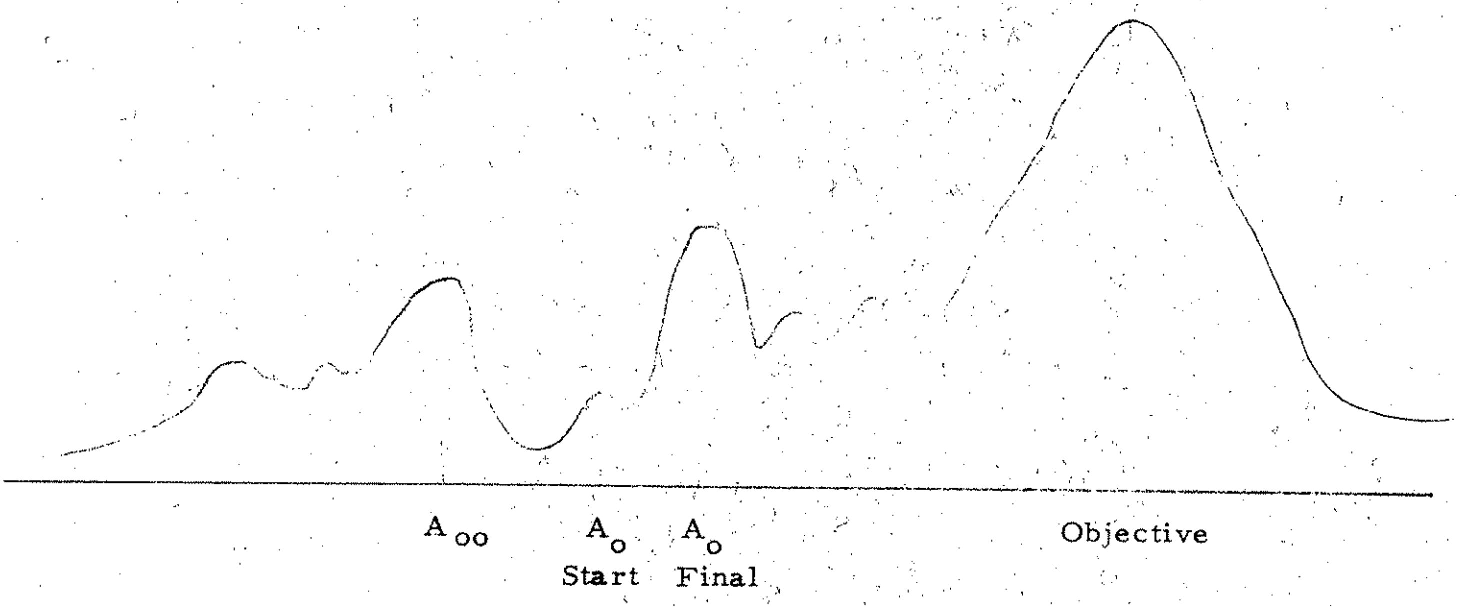


Figure 6.

A one-dimensional example showing base, A_{oo}, and link-bases A_o.

IV. Rotation of the Space.

The matrix M, defined below, represents an orthogonal transformation which rotates the space so that the vector, A, becomes the new n-th axis. This matrix is utilized in step 3 above.

Let

$$A(k_1),\ A(k_2),\ \ldots,\ A(k_{NZ})$$

be the sequence of non-zero components of A, and let

$$A(j_1),\ A(j_2),\ \ldots,\ Aj_Z)$$

be the components of A which are zero.

For $p = 1,\ Z$ put

$$M'(j_p - 1, i) = \begin{cases} 1 \text{ if } i = j_p \\ 0 \text{ if } i \neq j_p . \end{cases}$$

and for $p = 2,\ NZ$, put

$$M'(k_{p-1}, i) = \begin{cases} A(i), \text{ for } i = 1,\ k_p - 1 \\ -\displaystyle\sum_{j=1}^{k_p - 1} A^2(j) / A(k_p), \quad \text{for } i = k_p \\ 0, \text{ for } i = k_p + 1,\ n. \end{cases}$$

Also put

$$M'(n, i) = A(i), \text{ for } i = 1,\ n.$$

Normalize the matrix, M', to obtain the desired matrix, M:

$$M(k, i) = M'(k, i) \Big/ \left[\sum_{j=1}^{n} (M'(k, i))^2 \right]^{1/2} .$$

for $i = 1,\ M;\ k = 1,\ n$.

V. Experimental Results.

This method was coded for the CDC 1604 digital computer and tried out for several examples.

Figure 7 shows the results of a run where the learning set was the points shown in Figure 3. In this case the computer program worked very well to select the most optimum corridors and corresponding hyperplanes.

Other computer runs were made on handwritten characters with partial success. More longer runs are necessary to determine its effectiveness on such handwritten characters.

Data was prepared for runs with facial photographs but these were not made. It is for this application that the corridor-projection method would seem to have its most promise.

```
                  DICHOTOS
                                                   1st Test  2nd   3rd
*****RUN NUMBER =     1                                                    Categories
                                                      ↓       ↓     ↓          ↓
  N=   3  M=  40  MAXB=   3  MAXSTEPS(SQ)=  50  MAXSTEPS(IV)=  20  ESQ=   ,000
  X=    ,000000   ,000000   .000000      -       -     -
  X=    ,100000   ,100000   .000000      -       -     -
  X=   -,150000   ,000000  -,030000      -       -     -             0
  X=   -,100000  -,100000   .010000      -       -     -
  X=    ,135000   ,020000   .000000      -       -     -
  X=   1,000000  1,000000   .000000      -       -     +
  X=   1,100000  1,100000   .000000      -       -     +
  X=    ,850000  1,000000  -,030000      -       -     +             1
  X=    ,900000   ,900000   .010000      -       -     +
  X=   1,135000  1,020000   .000000      -       -     +
  X=    ,000000   ,200000  2,000000      +       -     -
  X=    ,100000   ,300000  2,000000      +       -     -
  X=   -,150000   ,200000  1,970000      +       -     -             4
  X=   -,100000   ,100000  2,010000      +       -     -
  X=    ,135000   ,220000  2,000000      +       -     -
  X=   1,000000  1,200000  2,000000      +       -     +
  X=   1,100000  1,300000  2,000000      +       -     +
  X=    ,850000  1,200000  1,970000      +       -     +
  X=    ,900000  1,100000  2,010000      +       -     +             5
  X=   1,350000  1,220000  2,000000      +       -     +
  X=   2,000000 -2,000000   .000000      -       +     -
  X=   2,100000 -1,900000   .000000      -       +     -
  X=   1,850000 -2,000000  -,030000      -       +     -             2
  X=   1,900000 -2,100000   .010000      -       +     -
  X=   2,135000 -1,980000   .000000      -       +     -
  X=   3,000000 -1,000000   .000000      -       +     +
  X=   3,100000  -,900000   .000000      -       +     +
  X=   2,850000 -1,000000  -,030000      -       +     +             3
  X=   2,900000 -1,100000   .010000      -       +     +
  X=   3,135000  -,980000   .000000      -       +     +
  X=   2,000000 -1,800000  2,000000      +       +     -
  X=   2,100000 -1,700000  2,000000      +       +     -
  X=   1,850000 -1,800000  1,970000      +       +     -             6
  X=   1,900000 -1,900000  2,010000      +       +     -
  X=   2,135000 -1,780000  2,000000      +       +     -
  X=   3,000000  -,800000  2,000000      +       +     +
  X=   3,100000  -,700000  2,000000      +       +     +
  X=   2,850000  -,800000  1,970000      +       +     +             8
  X=   2,900000  -,900000  2,010000      +       +     +
  X=   3,350000  -,780000  2,000000      +       +     +

  ALC =    1,000000  1,000000  1,000000
```

Figure 7

Results of a Successful Computer Run
Using the Corridor-Projection Method.